

IP Office 4.0

TAPI Link Entwicklerhandbuch

15-601035 Version 11 (22.01.2007)

Hinweis
Während redliche Bemühungen gemacht wurden, um sicherzustellen, dass die Information in diesem Dokument zur Zeit der Drucklegung vollständig und akkurat war, kann Avaya Inc. keine Haftung für irgendwelche Fehler übernehmen. Änderungen und Korrekturen an der Information in diesem Dokument können in zukünftigen Versionen eingefügt werden.

Haftungsausschluss für die Dokumentation
Avaya Inc. übernimmt keine Haftung für Änderungen, Zusätze oder Löschungen an der ursprünglich veröffentlichten Version dieser Dokumentation, außer wenn diese von Avaya durchgeführt wurden.

Haftungsausschluss für Links
Avaya Inc. übernimmt keine Verantwortung für den Inhalt oder die Zuverlässigkeit von durch Links verknüpften Websites und billigt nicht unbedingt die darin beschrieben oder angebotenen Produkte, Dienste oder Informationen. Wir können nicht garantieren, dass diese Links immer funktionieren werden, und wir haben keine Kontrolle über die Verfügbarkeit der gelinkten Seiten.

Lizenz
MIT DER INSTALLATION UND NUTZUNG DES PRODUKTS STIMMT DER ENDBENUTZER DEN IM FOLGENDEN DARGELEGTEN VERTRAGSBEDINGUNGEN UND DEN ALLGEMEINEN LIZENZVEREINBARUNGEN AUF DER AVAYA-WEBSITE UNTER http://support.avaya.com/LicenseInfo/ ZU ("ALLGEMEINE LIZENZVEREINBARUNG"). WENN SIE NICHT MIT DIESEN BEDINGUNGEN EINVERSTANDEN SIND, GEBEN SIE DAS PRODUKT INNERHALB VON ZEHN (10) TAGEN NACH LIEFERUNG AN DEN HÄNDLER ZURÜCK, UM EINE ERSTATTUNG ODER GUTSCHRIFT ZU ERHALTEN.

Avaya gewährt dem Endbenutzer eine Lizenz im Rahmen der nachfolgend beschriebenen Lizenztypen. Die anwendbare Anzahl von Lizenzen und Kapazitätseinheiten, für die eine Lizenz gewährt wird, beläuft sich auf eine Lizenz (1), sofern in der Dokumentation oder in anderen Dokumenten, die dem Endbenutzer zur Verfügung stehen, nicht eine andere Anzahl angegeben ist. "Designierter Prozessor" ist ein einzelner Standalone-Computer. "Server" ist ein designierter Prozessor, auf dem eine Softwareanwendung ausgeführt, auf die mehrere Benutzer zugreifen können. Bei "Software" handelt es sich um Computer-Programme im Objekt-Code, die ursprünglich von Avaya lizenziert sind und von Endbenutzern als Standalone-Produkte oder vorinstalliert auf Hardware verwendet werden. Bei "Hardware" handelt es sich um die standardmäßigen Hardware-Produkte, die ursprünglich von Avaya verkauft und vom Endbenutzer verwendet werden.

Lizenztypen: **Lizenz für designierte Systeme (DS).**
Der Endbenutzer darf jede Kopie der Software jeweils auf nur einem designierten Prozessor installieren und verwenden, sofern in der Dokumentation oder in anderen Dokumenten, die dem Endbenutzer vorliegen, nicht eine andere Anzahl angegeben ist. Avaya hat Anspruch darauf, dass der (die) designierte(n) Prozessor(en) durch Typ-, Seriennummer, Funktionsschlüssel, Standort oder andere spezifische Angaben identifiziert wird (werden); Diese Informationen kann der Endbenutzer Avaya auch mithilfe von elektronischen Mitteln mitteilen, die Avaya speziell zu diesem Zweck bereitstellt.

Copyright
Sofern nicht explizit anders angegeben, ist das Produkt durch Urheber- und andere Eigentumsrechte geschützt. Unerlaubte Vervielfältigung, Übertragung, und/oder Nutzung kann eine kriminelle sowie zivilrechtliche Verletzung unter zutreffenden Gesetzen darstellen.

Komponenten anderer Hersteller
Bestimmte in dem Produkt enthaltene Softwareprogramme oder Teile davon beinhalten Software, die unter Vertragsbedingungen anderer Hersteller ("Komponenten anderer Hersteller") vertrieben wird. Diese Bedingungen können die Nutzungsrechte bestimmter Teile des Produkts einschränken oder erweitern ("Bedingungen anderer Hersteller").

Avaya-Betrugsintervention
Wenn Sie meinen, dass Sie das Opfer von Gebührenhinterziehung sind und technische Hilfe oder Support benötigen, wenden Sie sich an die Betrugsinterventions-Hotline unseres technischen Servicecenters (+1-800-643-2353 für USA und Kanada. Vermutete Sicherheitsrisiken bei Avaya-Produkten sollten Avaya unter folgender E-Mail-Adresse mitgeteilt werden: securityalerts@avaya.com.

Weitere Support-Telefonnummern finden Sie auf der Support-Website von Avaya unter http://www.avaya.com/support.

Inhaltsverzeichnis

IP Office TAPI Link ... 1
Überblick ... 1
IP Office TAPI-Treiber ... 1
Erklärung ... 1
Referenzmaterial ... 2
Struktur dieses Dokuments ... 2
Installieren der TAPILink- und Wave-Treiber ... 3
Installieren der CTI TAPILink Pro- und Wave-Lizenzen ... 3
Konfiguration des TAPI-Treibers ... 3
Konfigurieren von IP Office für TAPI ... 5
Überblick ... 5
TAPI 2 ... 5
TAPI 3 ... 5

TAPI 2.x-Referenz ... 7
TAPI-Funktionen ... 7
TAPI-Funktionen ... 7
lineAddToConference ... 8
lineAnswer ... 8
lineBlindTransfer ... 8
lineCompleteTransfer ... 9
lineConfigDialog ... 9
lineClose ... 9
lineDeallocateCall ... 9
lineDevSpecific ... 10
lineDial ... 13
lineDrop ... 13
lineGenerateDigits ... 14
lineGenerateTone ... 14
lineGetAddressCaps ... 14
lineGetAddressID ... 15
lineGetAddressStatus ... 15
lineGetAppPriority ... 15
lineGetCallInfo ... 16
lineGetCallStatus ... 16
lineGetDevCaps ... 16
lineGetID ... 16
lineGetLineDevStatus ... 17
lineHandoff ... 19
lineHold ... 19
lineInitializeEx ... 19
lineMakeCall ... 20
lineMonitorDigits ... 20
lineMonitorTone ... 20
lineNegotiateAPIVersion ... 21
lineOpen ... 21
linePark ... 22
lineRedirect ... 22
lineRemoveFromConference ... 22
lineSetAppPriority ... 23
lineSetAppSpecific ... 23
lineSetCallPrivilege ... 23
lineSetStatusMessages ... 24
lineSetupTransfer ... 24
lineShutdown ... 24
lineSwapHold ... 25

lineUnhold 25
lineUnpark 25
TAPI-Strukturen 26
LINEADDRESSCAPS 26
LINEADDRESSSTATUS 31
LINECALLINFO 32
LINECALLPARAMS 33
LINECALLSTATUS 34
LINEDEVCAPS 35
TAPI-Ereignisse (Meldungen) 37
LINE_APPNEWCALL 37
LINE_CALLINFO 37
LINE_CALLSTATE 37
LINE_LINEDEVSTATE 37
LINE_DEVSPECIFIC 38
LINE_ADDRESSSTATE 38

TAPI 3.0-Referenz 39

TAPI 39
TAPI 39
ITTAPI 39
Address 40
Adresse 40
ITAddress 40
IEnumAddress 41
ITMediaSupport 41
Terminal 42
Call 42
Anrufen 42
ITCallInfo 42
ITBasicCallControl 43
ITCallStateEvent 45
ITCallNotificationEvent 46
ITCallInfoChangeEvent 46
CallHub 46

TAPI 3-Aufzählungstypen 47

CALL_STATE 47
CALLINFO_STRING 48
DISCONNECT_CODE 48
CALL_STATE_EVENT_CAUSE 49

IP Office Media Service Provider 51

Info zum MSP 51
Verwenden des MSP 51
Verwenden der gerätespezifischen Schnittstellen 51
ITACDAgent 52
ITGroup 52
ITDivert 53
ITPlay 54
IPOfficePrivateEvents 54
Verwenden der Media-Streaming-Funktionen des MSP 55

IP Office TAPI Link

Überblick

IP Office CTI Link ist verfügbar in Lite- und Pro-Versionen, die den Anwendungen zur Ausführungszeit Schnittstellen zur Verfügung stellen. Das Software Development Kit (SDK) beinhaltet eine Dokumentation der Lite- und Pro-Schnittstellen für die Softwareentwickler.

Lite und Pro umfassen das gleiche Programm. Die zusätzlichen Funktionen von IP Office CTI Link Pro werden aktiviert, wenn der CTI Link Pro-Lizenzschlüssel installiert ist. Weitere Informationen finden Sie im IP Office CTI Link-Installationshandbuch.

Dieses Dokument enthält Informationen, die für Softwareentwickler bei der Implementierung einer Anwendung hilfreich sind, die den IP Office TAPI-Dienstanbieter verwendet. Es wird vorausgesetzt, dass der Entwickler bereits mit TAPI vertraut ist. Des Weiteren ist beim Lesen dieses Dokuments der Zugriff auf die Microsoft Developer Network-Bibliothek (MSDN) von Nutzen, die eine vollständige TAPI-Dokumentation beinhaltet.

IP Office TAPI-Treiber

Die Architektur von Windows ermöglicht Entwicklern, Anwendungen unter Verwendung von standardmäßigen Anwendungsprogrammierschnittstellen (Application Programming Interface, API) zu implementieren, unabhängig von den verwendeten Telefonieanlagen. Hersteller von Telefonieanlagen stellen Telefonietreiber (Telephony Service Provider, TSP) bereit, die unter Windows installiert werden. Diese TSPs liefern die Verbindung zwischen TAPI und den Telefonieanlagen.

Der TAPI-Treiber für IP Office unterstützt alle TAPI-Versionen von 2.0 bis 3.0.

Erklärung

Avaya ist darum bemüht, dass die Releases des IP Office TAPI-Treibers mit älteren Versionen des Treibers rückwärts kompatibel sind, kann jedoch nicht gewährleisten, dass der Betrieb von IP Office mit den bereitgestellten Funktionen genau gleich abläuft. Aufgrund von Verbesserungen in IP Office ist es wahrscheinlich, dass sich die genaue Reihenfolge, die zeitlichen Abläufe und die Inhalte der TAPI-Ereignisse ändern. Wir empfehlen Entwicklern, ein ereignisgesteuertes Programmiermodell zu verwenden, damit durch solche Änderungen keine Nachteile entstehen.

Referenzmaterial

Wir empfehlen folgendes Referenzmaterial:

- MSDN/Plattform SDK
- Windows Telephony Programming (TAPI 1.x und 2.x).
- CTI Link-Installationshandbuch

Struktur dieses Dokuments

Dieses Dokument besteht aus folgenden Abschnitten:

- TAPI 2
    - TAPI-Funktionen (in alphabetischer Reihenfolge)
    - TAPI-Strukturen (in alphabetischer Reihenfolge)
    - TAPI-Ereignisse (in alphabetischer Reihenfolge)
- TAPI 3
- TAPI 3-Aufzählungstypen
- Media Service Provider
- Verlust und Wiederherstellung der Kommunikation

Installieren der TAPILink- und Wave-Treiber

Der IP Office TAPI-Dienstanbieter und der Wave-Treiber werden mit Hilfe der Benutzer-CD von IP Office installiert.

Weitere Informationen finden Sie im CTI Link-Installationshandbuch.

Installieren der CTI TAPILink Pro- und Wave-Lizenzen

Sie benötigen keine Lizenz, um den TAPI-Treiber zu verwenden, aber die Lizenz bietet die folgende zusätzliche Funktionalität:

- Drittanwendermodus
- Überwachung der ACD-Warteschlange
- aktivierte Funktion lineDevSpecific

Zur Verwendung der Wave-Funktionalität müssen Sie zusätzlich zur CTI Link Pro-Lizenz eine *Wave-Benutzerlizenz* für jeden Wave-Benutzer installieren.

Konfiguration des TAPI-Treibers

Der IP Office TAPI-Dienstanbieter kann im Einzelbenutzer- oder im Drittanwendermodus arbeiten. Für den Drittanwendermodus muss eine Lizenz erworben werden. Beachten Sie, dass Sie diese Option auch bei der nicht-lizenzierten Version (Lite) wählen können. Ohne Lizenz funktioniert sie jedoch nicht.

Einzelbenutzermodus bedeutet, dass mit der TAPI-Anwendung ein einzelnes Telefoniegerät kontrolliert und/oder überwacht werden kann. Drittanwendermodus bedeutet, dass mit der TAPI-Anwendung alle Telefoniegeräte in einer bestimmten IP Office-Einheit kontrolliert und/oder überwacht werden können.

Einzelbenutzermodus

Geben Sie die IP-Adresse der IP Office-Einheit in das Feld **IP-Adresse der Vermittlung** ein. Wählen Sie die Option **Einzelbenutzer**. Geben Sie den Benutzernamen und das Kennwort für die Nebenstelle ein, die von TAPI überwacht und/oder kontrolliert werden soll. Der Benutzername ist normalerweise der Name einer Person, die einer physischen Nebenstelle zugeordnet ist.

Drittanwendermodus

Geben Sie die IP-Adresse der IP Office-Einheit in das Feld **IP-Adresse der Vermittlung** ein. Wählen Sie die Option **Drittanwender**. Geben Sie das Kennwort für die Vermittlung ein. Hierbei handelt es sich um das gleiche Kennwort, das beim Laden der Vermittlungs-Konfiguration im Manager eingegeben wird.

Der Drittanwendermodus richtet standardmäßig eine TAPI-Leitung für jeden physischen Anschluss zum IP Office ein. Über die Kontrollkästchen für den Drittanwendermodus können zusätzliche Einheiten von TAPI überwacht und/oder gesteuert werden.

ACD-Warteschlangen

Das IP Office kann so konfiguriert werden, dass eingehende Anrufe, die an eine Gruppe interner Benutzer vermittelt werden, in eine Warteschlange gestellt werden. Wenn Ihr IP Office z. B. mit einer Gruppe von Call Center-Mitarbeitern konfiguriert wurde, können Sie einen eingehenden Anruf so lange in die Warteschlange stellen, bis ein Mitarbeiter den Anruf entgegennehmen kann.

Durch das Aktivieren des Kontrollkästchens ACD-Warteschlangen werden Leitungen zur Überwachung und/oder Kontrolle der wartenden Anrufe für eine Gruppe eingerichtet.

WAV-Benutzer

Bei Benutzern, deren Benutzername mit **TAPI** beginnt, handelt es sich um WAV-Benutzer. Die IP Office-Vermittlung versucht, Audio per Streaming an WAV-Benutzer weiterzuleiten, wenn diese an Anrufen beteiligt sind.

Für diese Audio-Streaming muss der IP Office-Wave-Treiber auf dem PC installiert sein. Der Treiber wird für jeden Benutzer einzeln lizenziert. Falls der Wave-Treiber nicht installiert ist, haben Sie eventuell noch das Kontrollkästchen WAV-Benutzer aktiviert und erhalten weiterhin WAV-Benutzerereignisse, ohne dass eine Lizenz erforderlich ist.

Für den TAPI-WAV-Audio-Stream wird ein IP Office-Datenkanal aus dem gleichen Pool wie für Voicemail-Ports verwendet. Die maximale Anzahl verfügbarer Datenkanäle für gleichzeitige Voicemail- und TAPI-WAV-Anrufe hängt vom Typ der IP Office-Steuereinheit ab. IP401 = 2, IP403 und Small Office Edition = 10, IP406 = 20, IP412 = 30.

Konfigurieren von IP Office für TAPI

In diesem Abschnitt wird die Konfiguration von IP Office mittels der Manager-Anwendung beschrieben. Wenn Ihre Anwendung Telefone überwacht aber nicht kontrolliert, ist keine Konfiguration notwendig.

Es gibt zwei Möglichkeiten, TAPI mit IP Office zu verwenden:

- Wenn Ihre Anwendung Telefone steuert, sollten Sie alle Benutzer, die gesteuert werden, als **Freisprechstation** konfigurieren Dadurch wird das Telefon des Benutzers in den Ruhezustand versetzt, wenn ein Anruf mittels TAPI aufgelegt wird. Ist diese Option nicht aktiviert, bleibt das Telefon im nicht-verbundenen Zustand, bis es manuell aufgelegt wird. Das Kontrollkästchen **Freisprechstation** befindet sich auf der Registerkarte Telefonie der Benutzereinstellungen im Manager.
- Wenn Sie möchten, dass Medienströme von einem bestimmten Benutzer gehandhabt werden (z.B. eine automatische Weiterleitung), erstellen Sie einen neuen Benutzer, dessen Name mit "TAPI:" beginnt. Damit kennzeichnen Sie ihn als Wave-Benutzer. Die Nummer dieses Benutzers sollte in einem Bereich liegen, der nicht mit vorhandenen Telefonnummern oder Gruppen in Konflikt tritt.

Überblick

Sie sollten alle TAPI-Anwendungen schließen, bevor Sie den Switch zurücksetzen. So kann der Telephony Service Provider (TSP) alle offenen Leitungen ordnungsgemäß schließen und es wird sichergestellt, dass der Switch und alle verbundenen TSPs einen einheitlichen Status aufweisen. Im Fall eines unerwarteten Kommunikationsverlusts (der Switch wird versehentlich heruntergefahren oder ein Netzwerkkabel wird versehentlich getrennt) erkennt der TSP, dass keine Verbindung mehr zum Switch besteht.

In diesem Zeitraum werden alle Aufrufe von TAPI-Funktionen abgelehnt, für die die Kommunikation zwischen TSP und Switch erforderlich ist. Die Zeit, die nach dem Kommunikationsverlust verstreicht, bis der TSP den Verlust erkennt, hängt von den TCP-Einstellungen auf dem Host und den TSP-internen Zeiteinstellungen ab. Es kann eine Verzögerung von bis zu zwei Minuten auftreten.

Nachdem der TSP erkannt hat, dass die Kommunikation mit dem Switch unterbrochen wurde, versucht er, die Verbindung wiederherzustellen. Bei erfolgreicher neuer Verbindung kann der Dienstanbieter die offenen Leitungen und Adressen in der Regel wiederherstellen. Dies ist auch dann der Fall, wenn der Kommunikationsverlust auf einen Neustart des Switch zurückzuführen ist.

Wie sich der Kommunikationsverlust für die TAPI-Anwendung darstellt, hängt von der verwendeten TAPI-Version ab, wie im Folgenden beschrieben.

TAPI 2

Wenn die Verbindung zwischen TSP und IP Office unterbrochen wird, wird über alle offenen Leitungen die Meldung LINEDEVSTATE_OUTOFSERVICE gesendet. Bei erfolgreicher Wiederherstellung der Verbindung wird für jede wiederhergestellte TAPI-Leitung die Meldung LINEDEVSTATE_INSERVICE gesendet.

TAPI 3

Wenn die Verbindung zwischen TSP und IP Office unterbrochen wird, wird für jede Adresse, die für den Erhalt solcher Ereignisse registriert ist, das Ereignis **ITAddressEvent** generiert. Dieses Ereignis gibt an, dass sich der Status der Adressen geändert hat. Der Status ändert sich in AS_OUTOFSERVICE. Bei erfolgreicher Wiederherstellung der Verbindung werden keine Ereignisse generiert. Es werden jedoch alle TAPI 3-Adressen wiederhergestellt.

TAPI 2.x-Referenz

TAPI-Funktionen

TAPI-Funktionen

In diesem Abschnitt werden die einzelnen Funktionen in TAPI 2.x beschrieben, die vom IP Office-TAPI-Treiber unterstützt werden. Es werden alle speziellen Vorgehensweisen und Beschränkungen der Funktionen bei der Verwendung mit IP Office erläutert.

- lineAddToConference
- lineAnswer
- lineBlindTransfer
- lineCompleteTransfer
- lineConfigDialog
- lineClose
- lineDeallocateCall
- lineDevSpecific
- lineDial
- lineDrop
- lineGenerateDigits
- lineGenerateTone
- lineGetAddressCaps
- lineGetAddressID
- lineGetAddressStatus
- lineGetAppPriority
- lineGetCallInfo
- lineGetCallStatus
- lineGetDevCaps
- lineGetID
- lineGetLineDevStatus
- lineHandoff
- lineHold
- lineInitializeEx
- lineMakeCall
- lineMonitorDigits
- lineMonitorTone
- lineNegotiateAPIVersion
- lineOpen
- linePark
- lineRedirect
- lineRemoveFromConference
- lineSetAppPriority
- lineSetAppSpecific
- lineSetCallPrivilege
- lineSetStatusMessages
- lineSetupTransfer
- lineShutdown
- lineSwapHold
- lineUnhold
- lineUnpark

lineAddToConference

Schaltet den Anruf in das Konferenzgespräch.

```
LONG
WINAPI
lineAddToConference(
HCALL hConfCall,
HCALL hConsultCall
);
```

lineAnswer

Beantwortet einen an der Anwendung wartenden Anruf.

```
LONG
WINAPI
lineAnswer(
HCALL hCall,
LPCSTR lpsUserUserInfo,
DWORD dwSize
);
```

Hinweis:

- **UserUserInfo** wird nicht unterstützt und somit ignoriert.

lineBlindTransfer

Diese Funktion dient zur Weiterleitung eines aktiven Anrufs an einen dritten Teilnehmer. Der Ländercode wird ignoriert.

```
LONG
WINAPI
lineBlindTransfer(
HCALL hCall,
LPCSTR lpszDestAddress,
DWORD dwCountryCode
);
```

lineCompleteTransfer

Mit dieser Funktion können Sie eine Vermittlung oder die Einleitung eines Konferenzgesprächs abschließen. Diese Funktion sollte eine Anruf-ID zurückgeben, gibt jedoch stets 0 aus.

```
LONG

WINAPI

lineCompleteTransfer(

HCALL hCall,
HCALL hConsultCall,
LPHCALL lphConfCall,
DWORD dwTransferMode
);
```

lineConfigDialog

Hiermit können Sie dasselbe Dialogfeld zur Konfiguration des TAPI-Dienstanbieters aufrufen wie unter **Systemsteuerung\Telefon- und Modemoptionen** (oder **Telefonie**). Der Parameter **lpszDeviceClass** wird ignoriert.

```
LONG

WINAPI

lineConfigDialog(

DWORD dwDeviceID,
HWND hwndOwner,
LPCSTR lpszDeviceClass
);
```

lineClose

Schließt eine Leitung. Rufen Sie diese Funktion auf, wenn Sie auf einer Leitung keine Anrufe mehr tätigen, empfangen oder überwachen möchten.

```
LONG

WINAPI

lineClose(

HLINE hLine
);
```

lineDeallocateCall

Hiermit können Sie die Zuordnung von Ressourcen zu einem Anruf aufheben. Diese Funktion sollte aufgerufen werden, wenn der Anruf inaktiv ist.

```
LONG

WINAPI

lineDeallocateCall(

HCALL hCall
);
```

lineDevSpecific

lineDevSpecific

TAPI bietet erweiterte Funktionalität durch die Funktion **lineDevSpecific**.

Beachten Sie, dass diese nur mit der lizenzierten Version des TAPI-Treibers verfügbar ist.

Mit Hilfe der Funktion **lineDevSpecific** wird ein Puffer unverändert an den TSP übermittelt, wo dieser wie gerätespezifische Befehle interpretiert wird. Es folgt eine Beschreibung der Befehlstypen:

```
LONG

WINAPI

lineDevSpecific(

HLINE hLine,
DWORD dwAddressID,
HCALL hCall,
LPVOID lpParams,
DWORD dwSize
);
```

Das Login-Protokoll

Um einen ACD-Agenten für die überwachte Leitung zu schalten, setzen Sie das erste Byte im Puffer auf **8**. Die folgenden Byte müssen eine Zeichenfolge für die Nebenstelle enthalten, die geschaltet wird. Mit dem folgenden Puffer wird der Agent 218 für die Leitung geschaltet, für die lineDevSpecific aufgerufen wird: -

```
unsigned char buf[6];

int len = 6;

buf[0] = 8; // Constant that means Login

buf[1] = '2';

buf[2] = '1';

buf[3] = '8';

buf[4] = 0; // Don't forget the null terminator
```

Abmelden

Sie können sich abmelden, indem Sie folgenden Puffer an die Funktion **DevSpecific** übergeben:

```
unsigned char buf[3];

int len = 3;

buf[0] = 9; // Constant that means Shortcode

buf[1] = 47; // Constant that means Log off

buf[2] = 0; // Don't forget the null terminator
```

Umleitungsziel

Um das Ziel für umgeleitete Anrufe festzulegen, senden Sie 9 im ersten Byte, 6 im zweiten Byte und in den folgenden Byte eine Zeichenfolge mit der Nebenstelle des Umleitungsziels. Wenn Sie beispielsweise die Nebenstelle 236 als Umleitungsziel festlegen möchten, senden Sie folgenden Puffer:

```
unsigned char buf[6];
int len = 6;
buf[0] = 9; // The first two bytes are devspecific constants
buf[1] = 6;
buf[2] = '2';
buf[3] = '3';
buf[4] = '6';
buf[5] = 0; // Don't forget the null terminator
```

Lampe "Nachricht wartet"

Bei einigen Telefonen leuchtet eine LED, wenn neue Voicemail-Nachrichten für den Benutzer vorliegen. Die Anzahl der neuen Nachrichten kann durch einen gerätespezifischen Befehl gesteuert werden. Bei 0 neuen Nachrichten leuchtet die LED nicht. Ab einer neuen Nachricht leuchtet die LED. Senden Sie den folgenden Puffer an **lineDevSpecific**:

```
unsigned char buf[21];
int len = 21;
buf[0] = 9; // Shortcode
buf[1] = 73; // Set MWL
sprintf(&(buffer[2]), ";Mailbox Msgs=%d", num);
// Where num is the number of messages
```

Hinweis:

- Beachten Sie, dass die LED auch vom IP Office-Server oder anderen IP Office-Anwendungen gesteuert werden kann.

Einstellungen für die Weiterleitung (Umleitung)

Die folgenden Konstanten helfen beim Ein- und Ausschalten von Umleitungsfunktionen:

```
const unsigned char ForwardAllOn = 0;
const unsigned char ForwardAllOff = 1;
const unsigned char ForwardBusyOn = 2;
const unsigned char ForwardBusyOff = 3;
const unsigned char ForwardNoAnswerOn = 4;
const unsigned char ForwardNoAnswerOff = 5;
const unsigned char DoNotDisturbOn = 7;
const unsigned char DoNotDisturbOff = 8;
```

Ein Puffer, der eine dieser Konstanten verwendet, sollte drei Byte lang sein und mit 9 beginnen. Mit dem folgenden Code wird die Leitung beispielsweise in den Modus **Nicht stören** geschaltet.

```
unsigned char buf[3];
int len = 3;
buf[0] = 9;
buf[1] = DoNotDisturbOn;
buf[2] = 0;
```

Sammelanschluss aktivieren/deaktivieren

Senden Sie folgenden Puffer, um die Zugehörigkeit des Benutzers zum Sammelanschluss mit der Nebenstelle *groupnum* zu aktivieren.

```
unsigned char buf[10];
int len = 10;
buf[0] = 9;
buf[1] = 76;
sprintf((char*)&buf[2], "%d", groupnum);
```

Senden Sie folgenden Puffer, um die Zugehörigkeit des Benutzers zum Sammelanschluss mit der Nebenstelle *groupnum* zu deaktivieren.

```
unsigned char buf[10];
int len = 10;
buf[0] = 9;
buf[1] = 77;
sprintf((char*)&buf[2], "%d", groupnum);
```

In beiden Fällen können Sie *alle* Zugehörigkeiten aktivieren bzw. deaktivieren, indem Sie die Nebenstellennummer durch 0 ersetzen (z.B. buf[2] = 0).

Hinweis:

- Beachten Sie, dass Sie die Zugehörigkeit eines Benutzers nur für Sammelanschlüsse aktivieren bzw. deaktivieren können, für die der Benutzer in Manager als Mitglied konfiguriert wurde.

Aufschalten

Senden Sie den folgenden Puffer, um sich in ein anderes Gespräch einzuschalten. Der Gesprächsteilnehmer, in dessen Gespräch Sie sich einschalten, wird durch die Ganzzahl *extnnum* dargestellt.

```
unsigned char buf[10];
int len = 10;
buf[0] = 9; // Shortcode
buf[1] = 83; // Intrude
sprintf((char*)&buf[2], "%d", extnnum);
```

Abhören

Senden Sie den folgenden Puffer, um ein anderes Gespräch abzuhören. Der Gesprächsteilnehmer, dessen Gespräch Sie abhören möchten, wird durch die Ganzzahl *extnnum* dargestellt.

```
unsigned char buf[10];
int len = 10;
buf[0] = 9; // Shortcode
buf[1] = 100; // Listen
sprintf((char*)&buf[2], "%d", extnnum);
```

lineDial

Mit dieser Funktion können Sie während eines Telefongesprächs eine Nummer wählen. Sie kann als Teil einer überwachten Weiterleitung verwendet werden (siehe lineSetupTransfer). Der Ländercode wird ignoriert.

```
LONG
WINAPI
lineDial(
HCALL hCall,
LPCSTR lpszDestAddress,
DWORD dwCountryCode
);
```

lineDrop

Mit dieser Funktion legen Sie auf bzw. beenden ein Gespräch. **UserUserInfo** wird nicht unterstützt und somit ignoriert.

```
LONG
WINAPI
lineDrop(
HCALL hCall,
LPCSTR lpsUserUserInfo,
DWORD dwSize
);
```

lineGenerateDigits

Rufen Sie diese Funktion auf, um bei einem Anruf DTMF-Ziffern zu generieren. Hierfür muss der Benutzer kein Wave-Benutzer sein und der Wave-Treiber muss nicht in den Anruf involviert sein. Wenn der Generierungsvorgang abgeschlossen ist, wird eine LINE_GENERATE-Meldung an die Anwendung gesendet. Der einzige unterstützte Wert für **dwDigitMode** ist LINEDIGITMODE_DTMF.

```
LONG

WINAPI

lineGenerateDigits(

HCALL hCall,
DWORD dwDigitMode,
LPCSTR lpszDigits,
DWORD dwDuration
);
```

lineGenerateTone

Mit dieser Funktion können Sie einen Signalton in der Leitung generieren. Bei der Leitung muss es sich um einen Wave-Benutzer handeln und der Wave-Treiber muss in den Anruf involviert sein. Der einzige unterstützte Wert für **dwToneMode** ist LINETONEMODE_BEEP. Da benutzerdefinierte Signaltöne nicht unterstützt werden, sollten Sie **dwNumTones** auf 0 setzen.

```
LONG

WINAPI

lineGenerateTone(

HCALL hCall,
DWORD dwToneMode,
DWORD dwDuration,
DWORD dwNumTones,
LPLINEGENERATETONE const lpTones
);
```

lineGetAddressCaps

Dient zum Abrufen der Telefoniefunktionen einer bestimmten Adresse für eine bestimmte Leitung. Die Funktionen werden in der Struktur LINEADDRESSCAPS zurückgegeben. Nähere Informationen hierzu finden Sie im Abschnitt zu TAPI-Strukturen unter **LINEADDRESSCAPS**.

IP Office-Leitungen verfügen stets über eine einzige Adresse.

```
LONG

WINAPI

lineGetAddressCaps(

HLINEAPP hLineApp,
DWORD dwDeviceID,
DWORD dwAddressID,
DWORD dwAPIVersion,
DWORD dwExtVersion,
LPLINEADDRESSCAPS lpAddressCaps
);
```

lineGetAddressID

Diese Funktion ermöglicht die Zuordnung der Telefonnummer (Adresse) einer Leitung zu ihrer *dwAddressID* im Bereich 0 bis zu der in LINEDEVCAPS zurückgegebenen Anzahl an Adressen minus 1. Wenn **dwNumAddresses** in LINEDEVCAPS den Wert 1 aufweist, gibt diese Funktion für den Wert DWORD, auf den **lpdwAddressID** verweist, stets 0 zurück.

```
LONG

WINAPI

lineGetAddressID(

HLINE hLine,
LPDWORD lpdwAddressID,
DWORD dwAddressMode,
LPCSTR lpsAddress,
DWORD dwSize
);
```

lineGetAddressStatus

Diese Funktion ermöglicht einer Anwendung, den aktuellen Status der angegebenen Adresse abzufragen. Nähere Informationen hierzu finden Sie im Abschnitt zu den TAPI-Strukturen unter **LINEADDRESSSTATUS**.

```
LONG

WINAPI

lineGetAddressStatus(

HLINE hLine,
DWORD dwAddressID,
LPLINEADDRESSSTATUS lpAddressStatus
);
```

lineGetAppPriority

Dient zum Abrufen der Anwendungspriorität.

```
LONG

WINAPI

lineGetAppPriority(

LPCSTR lpszAppFilename,
DWORD dwMediaMode,
LPLINEEXTENSIONID lpExtensionID,
DWORD dwRequestMode,
LPVARSTRING lpExtensionName,
LPDWORD lpdwPriority
);
```

lineGetCallInfo

Hiermit können feste Informationen über den angegebenen Anruf abgerufen werden. Nähere Informationen finden Sie in der Erläuterung zur Struktur LINECALLINFO.

```
LONG

WINAPI

lineGetCallInfo(

HCALL hCall,
LPLINECALLINFO lpCallInfo
);
```

lineGetCallStatus

Mit dieser Funktion wird eine LINECALLSTATUS-Struktur im Zusammenhang mit einem vorhandenen Anruf abgerufen. Nähere Informationen hierzu finden Sie im Abschnitt zu den TAPI-Strukturen unter **LINECALLSTATUS**.

```
LONG

WINAPI

lineGetCallStatus(

HCALL hCall,
LPLINECALLSTATUS lpCallStatus
);
```

lineGetDevCaps

Mit dieser Funktion können Sie die Struktur LINEDEVCAPS abrufen. Nähere Informationen hierzu finden Sie im Abschnitt zu den TAPI-Strukturen unter **LINEDEVCAPS**.

```
LONG

WINAPI

lineGetDevCaps(

HLINEAPP hLineApp,
DWORD dwDeviceID,
DWORD dwAPIVersion,
DWORD dwExtVersion,
LPLINEDEVCAPS lpLineDevCaps
);
```

lineGetID

Dient zum Abrufen der ID für eine Leitung, wenn **dwSelect** auf LINECALLSELECT_LINE gesetzt ist.

```
LONG

WINAPI

lineGetID(

HLINE hLine,
DWORD dwAddressID,
HCALL hCall,
DWORD dwSelect,
LPVARSTRING lpDeviceID,
LPCSTR lpszDeviceClass
);
```

lineGetLineDevStatus

Die Funktion **lineGetLineDevStatus** gibt einen gerätespezifischen Puffer zurück. Der gerätespezifische Puffer enthält folgende Informationen:

```
LONG

WINAPI

lineGetLineDevStatus(

HLINE hLine,
LPLINEDEVSTATUS lpLineDevStatus
);
```

Byte	Enthält	Anmerkung
0..n	Telefonnebenstelle	Die überwachte Nummer der Leitung als Zeichenfolge (z.B. "217").
n+1	0	Nullabschlusszeichen für die obige Zeichenfolge.
n+2	Rufweiterleitung bei Besetzt	1, wenn **Rufweiterleitung bei Besetzt** für das Telefon aktiviert ist, andernfalls 0.
n+3	Rufweiterleitung bei keine Antwort	1, wenn **Rufweiterleitung nach Zeit** für das Telefon aktiviert ist, andernfalls 0.
n+4	Rufweiterleitung sofort	1, wenn die sofortige Rufweiterleitung für das Telefon aktiviert ist.
n+5	Rufweiterleitung für Gruppenanrufe (Flag)	1, wenn die Rufweiterleitung für Sammelanschlussanrufe für das Telefon aktiviert ist.
n+6	Nicht stören	1, wenn **Nicht stören** für das Telefon aktiviert ist.
n+7	Sperre für ausgehende Anrufe (Flag)	1, wenn das Telefon für externe Anrufe gesperrt ist.
n+8	Anklopfen ein (Flag)	1, wenn die Anklopffunktion für dieses Telefon aktiviert ist.
n+9	Voicemail ein (Flag)	1, wenn die Voicemail-Funktion für dieses Telefon aktiviert ist.
n+10	Voicemail-Rückruf (Flag)	1, wenn die Voicemail-Rückruffunktion für dieses Telefon aktiviert ist.
n+11	Anzahl der gelesenen Voicemail-Nachrichten	Die Anzahl der gelesenen Nachrichten.
n+12	Anzahl ungelesener Voicemail-Nachrichten	Die Anzahl der neuen Voicemail-Nachrichten für den Benutzer.
n+13	Ruffolge extern (Anzahl)	Die Ruffolge für externe Anrufe.
n+14	Ruffolge intern (Anzahl)	Die Ruffolge für interne Anrufe.
n+15	Ruffolge bei Rückruf (Anzahl)	Die Ruffolge für Rückrufe.
n+16	Zeitüberschreitung bei keiner Antwort (Zeitspanne)	Gibt an, wie lange das Telefon klingelt (Sekunden), bevor die *keine Antwort*-Aktion ausgeführt wird, z. B. Weiterleitung oder Umleitung an Voicemail.
n+17	Abschlusszeit (Zeitspanne)	Der Zeitraum in Sekunden, in dem das Telefon nach einem Anruf keine Anrufe empfangen kann.

n+18	Aufschalten erlaubt (Flag)	1, wenn die Aufschaltung mit diesem Telefon möglich ist.
n+19	Aufschalten nicht erlaubt (Flag)	1, wenn eine Aufschaltung bei diesem Telefon nicht möglich ist.
n+20	Geheimnummer (Flag)	1, wenn dieser Benutzer nicht im internen Verzeichnis vorhanden ist.
n+21	Login erzwingen (Flag)	Der Benutzer ist beim Systemstart abgemeldet und muss sich somit anmelden.
n+22	Verrechnungscode erzwingen (Flag)	1, wenn für externe Anrufe mit diesem Telefon ein gültiger Verrechnungscode angegeben werden muss.
n+23	Login-Code (Flag)	1, wenn für diesen Benutzer ein Login-Code konfiguriert ist.
n+24	Systemtelefon (Flag)	1, wenn es sich um ein Systemtelefon handelt.
n+25	Abwesenheitsnachricht (ID)	Die ID der Abwesenheitsnachricht.
n+26	Abwesenheitsnachricht eingerichtet (Flag)	1, wenn die Abwesenheitsnachricht mit der im vorigen Feld angegebenen ID am Telefon angezeigt wird.
n+27	Voicemail als E-Mail (Modus)	1, wenn der Modus **Voicemail als E-Mail** aktiviert ist.
n+28..m	Nst	Die Benutzernebenstelle, die sich möglicherweise von der Telefonnebenstelle unterscheidet.
m+1	0	Nullabschlusszeichen für die obige Zeichenfolge (Nst).
m+2..p	Länderkennung	Die Länderkennung des Benutzers.
p+1	0	Das Nullabschlusszeichen für die Länderkennung.
p+2..q	Weiterleitungsziel	Die Nummer, die für dieses Telefon als Weiterleitungsziel festgelegt ist.
q+1	0	Das Nullabschlusszeichen für das oben angegebene Ziel.
q+2..r	Umleitungsnummer	Alle Anrufe werden an diese Nummer umgeleitet.
r+1	0	
r+2..s	Abwesenheitstext	Der für dieses Telefon definierte Abwesenheitstext.
s+1	0	
s+2..t	Nicht stören-Ausnahmeliste	Eine Liste mit Nummern, deren Anrufe für dieses Telefon durchgestellt werden, während es sich im Status **Nicht stören** befindet. Jede Nummer besteht aus einer Zeichenfolge mit Nullabschlusszeichen. Die letzte Nummer in der Liste endet mit zwei Nullen (das Feld **t+1** enthält die zweite Null). Wenn diese Liste leer ist, enthalten die Daten lediglich eine Null (im Feld **t+1**).
t+1	0	
t+2..u	Weiterleitungsnummer bei Besetzt	Die Nummer, an die Anrufe weitergeleitet werden, wenn dieses Telefon besetzt ist.
u+1	0	
u+2	Benutzerpriorität	Diese Priorität wird allen von diesem Benutzer getätigten Anrufen zugewiesen.

u+3	Gruppenmitgliedschaft	Dieses Byte enthält die Anzahl der Gruppen, für die der Benutzer zurzeit als Mitglied aktiviert ist.
u+4	Gruppen außerhalb des Zeitprofils	Anzahl der Gruppen, in denen der Benutzer Mitglied ist und die sich zurzeit außerhalb ihres Zeitprofils befinden.
u+5	Deaktivierte Gruppen	Anzahl der Gruppen, für die der Benutzer zurzeit deaktiviert ist.
u+6	Gruppen außer Betrieb	Anzahl der Gruppen, in denen der Benutzer Mitglied ist und die zurzeit außer Betrieb sind.
u+7	Gruppen im Nachtbetrieb	Anzahl der Gruppen, in denen der Benutzer Mitglied ist und die zurzeit im Nachtbetrieb sind.

lineHandoff

Mit der Funktion **lineHandoff** wird der angegebene Anruf an eine andere Anwendung übergeben.

```
LONG

WINAPI

lineHandoff(

HCALL hCall,
LPCSTR lpszFileName,
DWORD dwMediaMode
);
```

lineHold

Mit dieser Funktion wird ein aktiver Anruf gehalten.

```
LONG

WINAPI

lineHold(

HCALL hCall
);
```

lineInitializeEx

Dies ist die erste TAPI-Funktion, die zur Initialisierung von TAPI aufgerufen werden muss. Der Parameter **lpdwAPIVersion** muss mindestens auf **0x00020000** gesetzt werden.

```
LONG

WINAPI

lineInitializeEx(

LPHLINEAPP lphLineApp,
HINSTANCE hInstance,
LINECALLBACK lpfnCallback,
LPCSTR lpszFriendlyAppName,
LPDWORD lpdwNumDevs,
LPDWORD lpdwAPIVersion,
LPLINEINITIALIZEEXPARAMS lpLineInitializeExParams
);
```

lineMakeCall

Mit dieser Funktion wird ein Anruf getätigt. Nähere Informationen finden Sie am Ende des Kapitels zu den TAPI-Funktionen im Abschnitt zu den Anrufparametern.

```
LONG

WINAPI

lineMakeCall(

HLINE hLine,
LPHCALL lphCall,
LPCSTR lpszDestAddress,
DWORD dwCountryCode,
LPLINECALLPARAMS const lpCallParams
);
```

lineMonitorDigits

Rufen Sie diese Funktion auf, um die Erkennung von DTMF-Ziffern zu aktivieren. Diese Funktion kann nur verwendet werden, wenn der IP Office-Wave-Treiber in den Anruf involviert ist und der Benutzer ein Wave-Benutzer ist (siehe Abschnitt **Wave-Benutzer**). Die Erkennung erfolgt durch die Analyse von Medienstichproben im Wave-Treiber. Wenn ein DTMF-Ton erkannt wird, wird eine LINE_MONITORDIGITS-Meldung an die Anwendung gesendet. **dwDigitModes** kann auf LINEDIGITMODE_DTMF und/oder auf LINEDIGITMODE_DTMFEND gesetzt werden. Rufen Sie die Funktion **lineMonitorDigits** mit dem Wert 0 für **dwDigitMode** auf, um die Erkennung von DTMF-Ziffern abzubrechen.

```
LONG

WINAPI

lineMonitorDigits(

HCALL hCall,
DWORD dwDigitModes
);
```

lineMonitorTone

Damit diese Funktion ausgeführt werden kann, muss wie bei der vorigen Funktion der Wave-Treiber in den Anruf involviert sein. Darüber hinaus dient sie lediglich zur Erkennung von Momenten ohne Ton. Die Frequenzen in der Struktur LINEMONITORTONE, auf die **lpToneList** verweist, müssen alle auf 0 gesetzt werden. Bei Erkennung eines Moments ohne Ton wird eine LINE_MONITORTONE-Meldung an die Anwendung gesendet. Rufen Sie die Funktion **lineMonitorTone** mit dem Wert NULL für **lpToneList** auf, um die Erkennung von Momenten ohne Ton abzubrechen.

```
LONG

WINAPI

lineMonitorTone(

HCALL hCall,
LPLINEMONITORTONE const lpToneList,
DWORD dwNumEntries
);
```

lineNegotiateAPIVersion

Rufen Sie diese Funktion direkt nach **lineInitializeEx** auf, um sicherzustellen, dass die richtigen TAPI-Meldungen an Ihre Anwendung gesendet werden. Diese Funktion muss für jede von Ihrer Anwendung verwendete Leitung aufgerufen werden.

```
LONG

WINAPI

lineNegotiateAPIVersion(

HLINEAPP hLineApp,
DWORD dwDeviceID,
DWORD dwAPILowVersion,
DWORD dwAPIHighVersion,
LPDWORD lpdwAPIVersion,
LPLINEEXTENSIONID lpExtensionID
);
```

lineOpen

Diese Funktion dient zum Öffnen einer Leitung.

Bei ISDN/T1-Amtsleitungen muss **dwMediaModes** auf LINEMEDIAMODE_INTERACTIVEVOICE und bei analogen Amtsleitungen auf LINEMEDIAMODE_UNKNOWN gesetzt werden. Sie können auch beide Werte angeben, damit Anrufe beider Leitungstypen verarbeitet werden.

```
LONG

WINAPI

lineOpen(

HLINEAPP hLineApp,
DWORD dwDeviceID,
LPHLINE lphLine,
DWORD dwAPIVersion,
DWORD dwExtVersion,
DWORD dwCallbackInstance,
DWORD dwPrivileges,
DWORD dwMediaModes,
LPLINECALLPARAMS const lpCallParams
);
```

Hinweis:

- Wenn Sie versuchen, eine Leitung zu öffnen, die mit einem Wave-Benutzer verknüpft ist, in IP Office jedoch keine Wave-Benutzerlizenz installiert wurde, gibt die Funktion **lineOpen** den Wert LINEERR_RESOURCEUNAVAIL zurück. Eine Erläuterung zu Wave-Benutzern finden Sie unter **Wave-Benutzer**.

linePark

Mit dieser Funktion wird ein Anruf geparkt. Es wird nur der Modus LINEPARKMODE_DIRECTED unterstützt. Als Adressen können alphanumerische Zeichenfolgen verwendet werden. Da über ein Telefon jedoch nur Ziffern eingegeben werden können, möchten Sie die Adressen zum Parken von Anrufen möglicherweise auf numerische Zeichenfolgen beschränken. Die vier standardmäßigen Adressen in Phone Manager und eConsole lauten 1, 2, 3 und 4. Es wird empfohlen, diese Zahlen zu verwenden, wenn Sie geparkte Anrufe mit diesen Anwendungen unter Verwendung der Standardkonfiguration entparken möchten.

```
LONG

WINAPI

linePark(

HCALL hCall,
DWORD dwParkMode,
LPCSTR lpszDirAddress,
LPVARSTRING lpNonDirAddress
);
```

lineRedirect

The lineRedirect function redirects the specified offering call to the specified destination address. Der Ländercode wird ignoriert.

```
LONG

WINAPI

lineRedirect(

HCALL hCall,
LPCSTR lpszDestAddress,
DWORD dwCountryCode
);
```

lineRemoveFromConference

Diese Funktion dient zur Entfernung eines Anrufs aus einem Konferenzgespräch.

```
LONG

WINAPI

lineRemoveFromConference(

HCALL hCall
);
```

lineSetAppPriority

Rufen Sie diese Funktion auf, um Ihre Anwendungspriorität anzugeben.

```
LONG

WINAPI

lineSetAppPriority(

LPCSTR lpszAppFilename,
DWORD dwMediaMode,
LPLINEEXTENSIONID lpExtensionID,
DWORD dwRequestMode,
LPCSTR lpszExtensionName,
DWORD dwPriority
);
```

lineSetAppSpecific

Diese Funktion ermöglicht die Einrichtung des anwendungsspezifischen Feldes für den Informationsdatensatz des angegebenen Anrufs in einer Anwendung.

```
LONG

WINAPI

lineSetAppSpecific(

HCALL hCall,
DWORD dwAppSpecific
);
```

lineSetCallPrivilege

Rufen Sie diese Funktion auf, um die Anwendungs-Eigentumsrechte für einen bestimmten Anruf zu ändern.

```
LONG

WINAPI

lineSetCallPrivilege(

HCALL hCall,
DWORD dwCallPrivilege
);
```

lineSetStatusMessages

Mit dieser Funktion kann die Anwendung angeben, welche Benachrichtigungsmeldungen sie erfordert. In der Regel ist **dwLineStates**dwLineStates auf LINEDEVSTATE_ALL und **dwAddressStates** auf LINEADDRESSSTATE_ALL gesetzt.

```
LONG

WINAPI

lineSetStatusMessages(

HLINE hLine,
DWORD dwLineStates,
DWORD dwAddressStates
);
```

lineSetupTransfer

Diese Funktion dient zur Erstellung eines Rückfragerufs zur Durchführung einer überwachten Vermittlung. Der zu vermittelnde Anruf muss bereits vorhanden sein. Beim Aufruf dieser Funktion muss der Anruf entweder aktiv oder auf Halten gesetzt sein. Ein aktiver Anruf wird durch diese Funktion auf Halten gesetzt. Rufen Sie die Funktion **lineDial** auf, um den Teilnehmer anzurufen, an den der Anruf vermittelt werden soll. Schließen Sie die Vermittlung dann mit Hilfe der Funktion **lineCompleteTransfer** ab.

```
LONG

WINAPI

lineSetupTransfer(

HCALL hCall,
LPHCALL lphConsultCall,
LPLINECALLPARAMS const lpCallParams
);
```

lineShutdown

Hiermit beenden Sie die Verwendung der Funktionen für die TAPI-Leitung. In der Regel wird diese Funktion beim Herunterfahren Ihrer Anwendung aufgerufen.

```
LONG

WINAPI

lineShutdown(

HLINEAPP hLineApp
);
```

lineSwapHold

Mit dieser Funktion wird der aktuelle aktive Anruf auf Halten gesetzt und der gehaltene Anruf empfangen.

```
LONG

WINAPI

lineSwapHold(

HCALL hActiveCall,
HCALL hHeldCall
);
```

lineUnhold

Mit dieser Funktion wird ein gehaltener Anruf herangeholt. Wenn über die Leitung ein weiterer Gesprächspartner angerufen wird oder ein weiterer Anruf aktiv ist, wird bei Aufruf dieser Funktion der gestartete/aktive Anruf verworfen, bevor der gehaltene Anruf abgerufen wird.

```
LONG

WINAPI

lineUnhold(

HCALL hCall
);
```

lineUnpark

Diese Funktion dient zum Abrufen eines geparkten Anrufs. **dwAddressID** sollte auf **0** gesetzt sein, da IP Office-Leitungen lediglich über eine Adresse verfügen. **lpszDestAddress** sollte auf denselben Wert gesetzt werden, der auch zum Parken des Anrufs verwendet wurde (siehe **linePark**).

```
LONG

WINAPI

lineUnpark(

HLINE hLine,
DWORD dwAddressID,
LPHCALL lphCall,
LPCSTR lpszDestAddress
);
```

TAPI-Strukturen

LINEADDRESSCAPS

Diese Struktur wird von der Funktion **lineGetAddressCaps** zurückgegeben. In der folgenden Tabelle sind die Werte aufgeführt, die für Leitungen in Verbindung mit dem IP Office-TAPI-Treiber zurückgegeben werden.

Hinweis:

- Die Tabelle enthält nicht alle Details dieser Struktur. Eine ausführliche Erläuterung der möglichen Werte für LINEADDRESSCAPS finden Sie in der Dokumentation zu Microsoft TAPI.

Element	Beschreibung/Wert
dwLineDeviceID	Die ID der Leitung, mit der diese Adresse verbunden ist.
dwDevSpecificSize	Es werden keine zusätzlichen gerätespezifischen Informationen übergeben.
dwDevSpecificOffset	0
dwAddressSharing	LINEADDRESSSHARING_PRIVATE
dwAddressStates	0
dwCallInfoStates	Dient zur Rückgabe der möglichen Anrufinformationsstatus: LINECALLINFOSTATE_CALLID LINECALLINFOSTATE_RELATEDCALLID LINECALLINFOSTATE_NUMOWNERINCR LINECALLINFOSTATE_NUMOWNEDECR LINECALLINFOSTATE_NUMMONITORS LINECALLINFOSTATE_CALLERID LINECALLINFOSTATE_CALLEDID LINECALLINFOSTATE_REDIRECTIONID LINECALLINFOSTATE_REDIRECTINGID LINECALLINFOSTATE_DISPLAY LINECALLINFOSTATE_MONITORMODES LINECALLINFOSTATE_CALLDATA
dwCallerIDFlags	Dient zur Rückgabe der möglichen Markierungen für die Rufnummer des Anrufers: LINECALLPARTYID_BLOCKED LINECALLPARTYID_OUTOFAREA LINECALLPARTYID_NAME LINECALLPARTYID_ADDRESS LINECALLPARTYID_UNKNOWN LINECALLPARTYID_UNAVAIL
dwCalledIDFlags	Dient zur Rückgabe der möglichen Markierungen für die Rufnummer des angerufenen Teilnehmers: LINECALLPARTYID_BLOCKED LINECALLPARTYID_OUTOFAREA LINECALLPARTYID_NAME

	LINECALLPARTYID_ADDRESS LINECALLPARTYID_UNKNOWN LINECALLPARTYID_UNAVAIL
dwConnectedIDFlags	Dient zur Rückgabe der möglichen Verbindungs-ID-Markierungen: LINECALLPARTYID_NAME LINECALLPARTYID_UNKNOWN LINECALLPARTYID_UNAVAIL
dwRedirectionIDFlags	Dient zur Rückgabe der möglichen Markierungen für die Rufnummer des Umleitungsziels: LINECALLPARTYID_BLOCKED LINECALLPARTYID_OUTOFAREA LINECALLPARTYID_NAME LINECALLPARTYID_ADDRESS LINECALLPARTYID_UNKNOWN LINECALLPARTYID_UNAVAIL
dwRedirectingIDFlags	Dient zur Rückgabe der möglichen Markierungen für die Rufnummer, von der die Umleitung ausgeht: LINECALLPARTYID_BLOCKED LINECALLPARTYID_OUTOFAREA LINECALLPARTYID_NAME LINECALLPARTYID_ADDRESS LINECALLPARTYID_UNKNOWN LINECALLPARTYID_UNAVAIL
dwCallStates	Dient zur Rückgabe der möglichen Anrufstatus: LINECALLSTATE_IDLE (Anruf nicht mehr vorhanden) LINECALLSTATE_OFFERING (neuen Anruf empfangen) LINECALLSTATE_ACCEPTED (Anruf wurde von einer Anwendung angefordert) LINECALLSTATE_DIALTONE (Anrufer hört Wählton) LINECALLSTATE_DIALING (Vermittlung empfängt Wählinformationen) LINECALLSTATE_RINGBACK (Anrufer hört das Klingelzeichen) LINECALLSTATE_BUSY (Anrufer hört das Besetztzeichen) LINECALLSTATE_CONNECTED (Verbindung vollständig hergestellt) LINECALLSTATE_PROCEEDING (Wählvorgang abgeschlossen, Anruf noch nicht durchgestellt) LINECALLSTATE_ONHOLD (Anruf wurde auf Halten gesetzt) LINECALLSTATE_CONFERENCED (Anruf für eine Konferenz) LINECALLSTATE_ONHOLDPENDCONF (Anruf vor Konferenzschaltung auf Halten gesetzt) LINECALLSTATE_ONHOLDPENDTRANSFER (Anruf vor

	Vermittlung auf Halten gesetzt) LINECALLSTATE_DISCONNECTED (Verbindung wurde vom anderen Gesprächsteilnehmer getrennt) LINECALLSTATE_UNKNOWN (Anrufstatus unbekannt)
dwDialToneModes	Dient zur Rückgabe des möglichen Wähltonmodus: LINEDIALTONEMODE_UNAVAIL
dwBusyModes	Dient zur Rückgabe der möglichen Besetzt-Modi: LINEBUSYMODE_UNAVAIL
dwSpecialInfo	Dient zur Rückgabe der möglichen Spezialinformationen aus LINESPECIALINFO_UNAVAIL
dwDisconnectModes	Dient zur Rückgabe der möglichen Trennen-Modi: LINEDISCONNECTMODE_NORMAL LINEDISCONNECTMODE_REJECT LINEDISCONNECTMODE_PICKUP LINEDISCONNECTMODE_FORWARDED LINEDISCONNECTMODE_BUSY LINEDISCONNECTMODE_NOANSWER LINEDISCONNECTMODE_BADADDRESS LINEDISCONNECTMODE_UNREACHABLE LINEDISCONNECTMODE_CONGESTION LINEDISCONNECTMODE_INCOMPATIBLE LINEDISCONNECTMODE_UNAVAIL LINEDISCONNECTMODE_NODIALTONE LINEDISCONNECTMODE_QOSUNAVAIL LINEDISCONNECTMODE_BLOCKED LINEDISCONNECTMODE_DONOTDISTURB
dwMaxNumActiveCalls	Die maximale Anzahl an aktiven Anrufen: 1
dwMaxNumOnHoldCalls	Die maximale Anzahl an gehaltenen Anrufen: 9
dwMaxNumOnHoldPendingCalls	Die maximale Anzahl an ausstehenden gehaltenen Anrufen: 9
dwMaxNumConference	Die maximale Anzahl an Konferenzgesprächen: 9
dwMaxNumTransConf	Die maximale Anzahl an vermittelten Konferenzgesprächen: 9
dwAddrCapFlags	Dient zur Rückgabe der möglichen Markierungen für den Adresswert: LINEADDRCAPFLAGS_FWDNUMRINGS LINEADDRCAPFLAGS_DIALED LINEADDRCAPFLAGS_TRANSFERHELD LINEADDRCAPFLAGS_TRANSFERMAKE LINEADDRCAPFLAGS_CONFERENCEHELD LINEADDRCAPFLAGS_CONFERENCEMAKE LINEADDRCAPFLAGS_FWDSTATUSVALID

dwCallFeatures	Dient zur Rückgabe der möglichen Anruffunktionen: LINECALLFEATURE_ADDTOCONF LINECALLFEATURE_ANSWER LINECALLFEATURE_BLINDTRANSFER LINECALLFEATURE_COMPLETETRANSF LINECALLFEATURE_DIAL LINECALLFEATURE_DROP LINECALLFEATURE_GENERATEDIGITS LINECALLFEATURE_HOLD LINECALLFEATURE_PARK LINECALLFEATURE_REDIRECT LINECALLFEATURE_REMOVEFROMCONF LINECALLFEATURE_SETUPTRANSFER LINECALLFEATURE_SWAPHOLD LINECALLFEATURE_UNHOLD LINECALLFEATURE_SETCALLDATA
dwRemoveFromConfCaps	Dient zur Rückgabe der möglichen Werte für die Entfernung aus einer Konferenz: LINEREMOVEFROMCONF_ANY
dwRemoveFromConfState	Dient zur Rückgabe des möglichen Status für die Entfernung aus einer Konferenz: LINECALLSTATE_ONHOLD
dwTransferModes	Dient zur Rückgabe der möglichen Vermittlungsmodi: LINETRANSFERMODE_TRANSFER LINETRANSFERMODE_CONFERENCE
dwParkModes	Dient zur Rückgabe des möglichen Parkmodus LINEPARKMODE_DIRECTED
dwForwardModes	Dient zur Rückgabe der möglichen Weiterleitungsmodi: LINE FORWARDMODE_UNCOND LINE FORWARDMODE_UNCONDEXTERNAL LINE FORWARDMODE_UNCONDSPECIFIC LINE FORWARDMODE_BUSY LINE FORWARDMODE_BUSYINTERNAL LINE FORWARDMODE_BUSYEXTERNAL LINE FORWARDMODE_BUSYSPECIFIC LINE FORWARDMODE_NOANSW LINE FORWARDMODE_NOANSWINTERNAL LINE FORWARDMODE_NOANSWEXTERNAL LINE FORWARDMODE_NOANSWSPECIFIC LINE FORWARDMODE_BUSYNA LINE FORWARDMODE_BUSYNAINTERNAL LINE FORWARDMODE_BUSYNAEXTERNAL LINE FORWARDMODE_BUSYNASPECIFIC
dwMaxForwardEntries	Die maximale Anzahl an weitergeleiteten Einträgen: 10

dwMaxSpecificEntries	Die maximale Anzahl an spezifischen Einträgen: 10
dwMinFwdNumRings	Die minimale Anzahl an Klingeltönen bei der Weiterleitung: 1
dwMaxFwdNumRings	Die maximale Anzahl an Klingeltönen bei der Weiterleitung: 99
dwMaxCallCompletions	0
dwCallCompletionConds	0
dwCallCompletionModes	0
dwNumCompletionMessages	0
dwCompletionMsgTextEntrySize	0
dwCompletionMsgTextSize	0
dwCompletionMsgTextOffset	0
dwAddressFeatures	Dient zur Rückgabe der möglichen Adressfunktionen: LINEADDRFEATURE_FORWARD LINEADDRFEATURE_MAKECALL LINEADDRFEATURE_SETUPCONF LINEADDRFEATURE_UNPARK LINEADDRFEATURE_FORWARDFWD LINEADDRFEATURE_FORWARDDND
dwPredictiveAutoTransferStates	0
dwNumCallTreatments	0
dwCallTreatmentListSize	0
dwCallTreatmentListOffset	0
dwDeviceClassesSize	0
dwDeviceClassesOffset	0
dwMaxCallDataSize	Die maximale Anrufdatengröße: 127
dwCallFeatures2	0
dwMaxNoAnswerTimeout	0
dwConnectedModes	0
dwOfferingModes	0
dwAvailableMediaModes	0

LINEADDRESSSTATUS

Diese Struktur wird von der Funktion **lineGetAddressStatus** zurückgegeben.

Hinweis:

- Die Tabelle enthält nicht alle Details dieser Struktur. Eine ausführliche Erläuterung der möglichen Werte für LINEADDRESSSTATUS finden Sie in der Dokumentation zu Microsoft TAPI.

Element	Beschreibung/Wert
dwNumInUse;	Immer 1
dwNumActiveCalls;	Anzahl der aktiven Anrufe
dwNumOnHoldCalls;	Immer 0
dwNumOnHoldPendCalls;	Immer 0
dwAddressFeatures;	Dient zur Angabe der möglichen Funktionen: LINEADDRFEATURE_MAKECALL LINEADDRFEATURE_SETUPCONF LINEADDRFEATURE_UNPARK
dwNumRingsNoAnswer;	5
dwForwardNumEntries;	Immer 0
dwForwardSize; /	Immer 0
dwForwardOffset;	Immer 0
dwTerminalModesSize;	Immer 0
dwTerminalModesOffset;	Immer 0
dwDevSpecificSize;	Immer 0

LINECALLINFO

Diese Struktur wird von der Funktion **lineGetCallInfo** zurückgegeben.

Hinweis:

- Die Tabelle enthält nicht alle Details dieser Struktur. Eine ausführliche Erläuterung der möglichen Werte für LINECALLINFO finden Sie in der Dokumentation zu Microsoft TAPI.

Element	Beschreibung/Wert
dwAddressID	Immer 0
dwBearerMode	Dient zur Rückgabe des möglichen Dienstmodus: LINEBEARERMODE_VOICE
dwRate	64000
dwMediaMode	Dient zur Rückgabe des möglichen Medienmodus: LINEMEDIAMODE_INTERACTIVEVOICE
dwAppSpecific	Einstellung durch Anwendung
dwCallID	Anrufer-ID
dwCallParamFlags	Dient zur Rückgabe der möglichen Markierung für Anrufparameter: LINECALLPARAMFLAGS_IDLE
dwCallStates	Dient zur Rückgabe der möglichen Anrufstatus: LINECALLSTATE_IDLE LINECALLSTATE_OFFERING LINECALLSTATE_DIALTONE LINECALLSTATE_DIALING LINECALLSTATE_RINGBACK LINECALLSTATE_BUSY LINECALLSTATE_CONNECTED LINECALLSTATE_PROCEEDING LINECALLSTATE_ONHOLD LINECALLSTATE_CONFERENCED LINECALLSTATE_ONHOLDPENDCONF LINECALLSTATE_ONHOLDPENDTRANSFER LINECALLSTATE_DISCONNECTED LINECALLSTATE_UNKNOWN
dwMonitorMediaModes	0
dwCountryCode	0
dwTrunk	0xFFFFFFFF
dwCommentSize	0
dwCommentOffset	0
dwUserUserInfoSize	0

dwUserUserInfoOffset	0
dwHighLevelCompSize	0
dwHighLevelCompOffset	0
dwLowLevelCompSize	0
dwLowLevelCompOffset	0
dwChargingInfoSize	0
dwChargingInfoOffset	0
dwTerminalModesSize	0
dwTerminalModesOffset	0
dwCallDataOffset	0
dwSendingFlowspecSize	0
dwSendingFlowspecOffset	0
dwReceivingFlowspecSize	0
dwReceivingFlowspecOffset	0
dwCallerIDAddressType	0 – nur gültig ab TAPI Version 3.0
dwCalledIDAddressType	0 – nur gültig ab TAPI Version 3.0
dwConnectedIDAddressType	0 – nur gültig ab TAPI Version 3.0
dwRedirectionIDAddressType	0 – nur gültig ab TAPI Version 3.0
dwRedirectingIDAddressType	0 – nur gültig ab TAPI Version 3.0

LINECALLPARAMS

Folgende Parameter der Struktur LINECALLPARAMS können an **lineMakeCall** und **lineSetupTransfer** übergeben werden:

Hinweis:

- Die Tabelle enthält nicht alle Details dieser Struktur. Eine ausführliche Erläuterung der möglichen Werte für LINECALLPARAMS finden Sie in der Dokumentation zu Microsoft TAPI.

Element	Beschreibung/Wert
dwCallParamFlags	Setzen Sie diese Option für die normale Verwendung auf 0. Wenn Sie die Rufnummer des Anrufers verbergen möchten, geben Sie LINEBEARERMODE_VOICE ein.
dwCalledPartyOffset	Kann zur Einstellung der Rufnummer des angerufenen Teilnehmers verwendet werden.
dwCallingPartyIDOffset	Kann zur Einstellung der Rufnummer des Anrufers verwendet werden.

LINECALLSTATUS

Diese Struktur wird von der Funktion **lineGetCallStatus** zurückgegeben.

Hinweis:

- Die Tabelle enthält nicht alle Details dieser Struktur. Eine ausführliche Erläuterung der möglichen Werte für LINECALLSTATUS finden Sie in der Dokumentation zu Microsoft TAPI.

Element	Beschreibung/Wert
dwCallState	Gibt einen der folgenden Status zurück: LINECALLSTATE_IDLE (Anruf nicht mehr vorhanden) LINECALLSTATE_OFFERING (neuen Anruf empfangen) LINECALLSTATE_ACCEPTED (Anruf wurde von einer Anwendung angefordert) LINECALLSTATE_DIALTONE (Anrufer hört Wählton) LINECALLSTATE_DIALING (Vermittlung empfängt Wählinformationen) LINECALLSTATE_RINGBACK (Anrufer hört das Klingelzeichen) LINECALLSTATE_BUSY (Anrufer hört das Besetztzeichen) LINECALLSTATE_CONNECTED (Verbindung vollständig hergestellt) LINECALLSTATE_PROCEEDING (Wählvorgang abgeschlossen, Anruf noch nicht durchgestellt) LINECALLSTATE_ONHOLD (Anruf wurde auf Halten gesetzt) LINECALLSTATE_CONFERENCED (Anruf für eine Konferenz) LINECALLSTATE_ONHOLDPENDCONF (Anruf vor Konferenzschaltung auf Halten gesetzt) LINECALLSTATE_ONHOLDPENDTRANSFER (Anruf vor Vermittlung auf Halten gesetzt) LINECALLSTATE_DISCONNECTED (Verbindung wurde vom anderen Gesprächsteilnehmer getrennt) LINECALLSTATE_UNKNOWN (Anrufstatus unbekannt)
dwCallStateMode	immer 0
dwCallPrivilege	Die Anwendungsberechtigung für diesen Anruf
dwCallFeatures	Die für den durch **dwCallState** angegebenen Anrufstatus verfügbaren Anruffunktionen. TAPI gibt alle möglichen Funktionen an, es können jedoch nur die in der Struktur LINEADDRESSCAPS unter **dwCallFeatures** aufgeführten Funktionen verwendet werden.
dwDevSpecificSize	0
dwDevSpecificOffset	0
dwCallFeatures2	0
tStateEntryTime	Nullen

LINEDEVCAPS

Diese Struktur wird von der Funktion **lineGetDevCaps** zurückgegeben. In der folgenden Tabelle sind die Werte aufgeführt, die für Leitungen in Verbindung mit dem IP Office-TAPI-Treiber zurückgegeben werden.

Hinweis:

- Die Tabelle enthält nicht alle Details dieser Struktur. Eine ausführliche Erläuterung der möglichen Werte für LINEDEVCAPS finden Sie in der Dokumentation zu Microsoft TAPI.

Element	Beschreibung/Wert
dwProviderInfoSize	Name des Providers, d.h. Name des TSP
dwSwitchInfoSize	0
dwPermanentLineID	Eindeutige, von Windows zugewiesene ID
dwLineNameSize	Leitungsname
dwStringFormat	Dient zur Rückgabe des Zeichenfolgenformats: STRINGFORMAT_ASCII
dwAddressModes	Dient zur Rückgabe des Adressmodus: LINEADDRESSMODE_ADDRESSID
dwNumAddresses	1
dwBearerModes	Dient zur Rückgabe der möglichen Dienstmodi: LINEBEARERMODE_VOICE LINEBEARERMODE_SPEECH
dwMaxRate	0
dwMediaModes	Dient zur Rückgabe des Medienmodus: LINEMEDIAMODE_INTERACTIVEVOICE
dwGenerateToneModes	Dient zur Rückgabe des Ton-Generierungs-Modus: LINETONEMODE_BEEP
dwGenerateToneMaxNumFreq	0
dwMonitorToneMaxNumFreq	1
dwMonitorToneMaxNumEntries	1
dwGatherDigitsMinTimeout	0
dwGatherDigitsMaxTimeout	0
dwMedCtlDigitMaxListSize	0
dwMedCtlMediaMaxListSize	0
dwMedCtlToneMaxListSize	0
dwMedCtlCallStateMaxListSize	0
dwDevCapFlags	Dient zur Rückgabe der Markierungen für die Gerätewerte: LINEDEVCAPFLAGS_CLOSEDROP LINEDEVCAPFLAGS_DIALBILLING

	LINEDEVCAPFLAGS_DIALQUIET LINEDEVCAPFLAGS_DIALDUALTONE
dwMaxNumActiveCalls	9
dwAnswerMode	Dient zur Rückgabe des Antwortmodus: LINEANSWERMODE_NONE
dwRingModes	1
dwLineStates	Dient zur Rückgabe des Leitungsstatus: LINEDEVSTATE_RINGING LINEDEVSTATE_CONNECTED LINEDEVSTATE_DISCONNECTED LINEDEVSTATE_INSERVICE LINEDEVSTATE_OUTOFSERVICE LINEDEVSTATE_OPEN LINEDEVSTATE_CLOSE LINEDEVSTATE_REINIT LINEDEVSTATE_TRANSLATECHNGE LINEDEVSTATE_REMOVED
dwUUIAcceptSize	0
dwUUIAnswerSize	100
dwUUIMakeCallSize	100
dwUUIDropSize	100
dwUUISendUserUserInfoSize	100
dwUUICallInfoSize	Anrufdatengröße Benutzer zu Benutzer: 100
dwNumTerminals	0
dwTerminalCapsSize	0
dwTerminalCapsOffset	0
dwTerminalTextEntrySize	0
dwTerminalTextSize	0
dwTerminalTextOffset	0
dwDevSpecificSize	0
dwDevSpecificOffset	0
dwLineFeatures	Dient zur Rückgabe der Leitungsfunktion: LINEFEATURE_MAKECALL
dwSettableDevStatus	0
dwDeviceClassesSize	tapi\line
PermanentLineGuide	Relevant ab TAPI Version 2.2.
dwAddressTypes	Relevant ab TAPI Version 3.0.
ProtocolGuide	Relevant ab TAPI Version 3.0.
dwAvailableTracking	Relevant ab TAPI Version 3.0.

TAPI-Ereignisse (Meldungen)

LINE_APPNEWCALL

Ein neuer Anruf wurde erstellt.

LINE_CALLINFO

Informationen in der Struktur LINECALLINFO haben sich geändert.

LINE_CALLSTATE

Der Status des Anrufs hat sich geändert. Eine Liste der unterstützten Status finden Sie in der Struktur LINEADDRESSCAPS unter **dwCallStates**.

LINE_LINEDEVSTATE

Der Status der Leitung hat sich geändert. Für den zweiten Parameter bestehen folgende Möglichkeiten:

- LINEDEVSTATE_DEVSPECIFIC: Gerätespezifische Informationen haben sich geändert.
- LINEDEVSTATE_CONNECTED, LINEDEVSTATE_DISCONNECTED: Der Verbindungsstatus der Leitung hat sich geändert.
- LINEDEVSTATE_OUTOFSERVICE: Die Kommunikation zwischen TSP und Vermittlung wurde unterbrochen. Diese Leitung ist nun außer Betrieb.
- LINEDEVSTATE_INSERVICE: Die Kommunikation zwischen TSP und Vermittlung wurde unterbrochen, die Verbindung konnte jedoch wiederhergestellt werden und die Leitung ist wieder in Betrieb.
- LINEDEVSTATE_RINGING: Die Vermittlung hat festgestellt, dass das Telefon des anrufenden Teilnehmers klingelt.

LINE_DEVSPECIFIC

Informiert die Anwendung über gerätespezifische Ereignisse in einer Leitung, Adresse oder einem Anruf. Die Anwendung wird durch diese Meldung aufgefordert, die Funktion **lineGetLineDevStatus** aufzurufen und den gerätespezifischen Puffer auf Änderungen zu überprüfen.

LINE_ADDRESSSTATE

Der Status einer Adresse wurde in einer Leitung geändert, die zurzeit von der Anwendung geöffnet ist.

TAPI 3.0-Referenz

TAPI

TAPI

Das TAPI-Objekt wird von **CoCreateInstance** erstellt. Alle anderen Objekte in TAPI 3.0 werden von TAPI 3.0 erstellt.

ITTAPI

Die Schnittstelle **ITTAPI** dient als Basisschnittstelle für das TAPI-Objekt.

Initialize

Dies ist die erste TAPI-Funktion, die zur Initialisierung von TAPI aufgerufen werden muss.

```
HRESULT
Initialize();
```

Shutdown

Hiermit wird eine TAPI-Sitzung geschlossen. In der Regel wird diese Funktion beim Herunterfahren Ihrer Anwendung aufgerufen.

```
HRESULT
Shutdown();
```

EnumerateAddresses

Diese Methode listet die aktuell verfügbaren Adressen auf.

```
HRESULT
EnumerateAddresses ( IEnumAddress **ppEnumAddress );
```

RegisterCallNotifications

Hiermit wird festgelegt, welche neuen Anrufbenachrichtigungen eine Anwendung empfängt. Die Anwendung muss diese Methode für jede Adresse aufrufen und die unterstützten Medientypen sowie die angeforderten Berechtigungen müssen angegeben werden.

```
HRESULT RegisterCallNotifications(
ITAddress *pAddress,
VARIANT_BOOLfMonitor,
VARIANT_BOOLfOwner,
long lMediaTypes,
long lCallbackInstance,
long *plRegister
);
```

put_EventFilter

Die Methode **put_EventFilter** stellt die Ereignis-Filtermaske ein.

```
HRESULT
put_EventFilter ( long lFilterMask);
```

Address

Adresse

Das Adressobjekt (Address) stellt eine Einheit dar, die Anrufe senden und empfangen kann.

ITAddress

Diese Schnittstelle dient als Basisschnittstelle für das Objekt **Address**.

get_AddressName

Ruft den Anzeigename der Adresse ab.

```
HRESULT
get_AddressName (BSTR *ppName );
```

get_DialableAddress

Die Methode **get_DialableAddress** ruft den Wert **BSTR** ab, der zur Herstellung einer Verbindung mit dieser Adresse verwendet werden kann.

```
HRESULT
get_DialableAddress (
BSTR *pDialableAddress
);
```

get_ServiceProviderName

Die Methode **get_ServiceProviderName** ruft den Namen des Telefonie-Dienstanbieters (Telephony Service Provider, TSP) ab, der diese Adresse unterstützt. Beispiel: **Unimdm.tsp** für den Dienstanbieter Unimodem oder **H323.tsp** für den Dienstanbieter H323.

```
HRESULT
get_ServiceProviderName (
BSTR *ppName
);
```

CreateCall

Die Methode **CreateCall** erstellt ein neues Call-Objekt, das für einen ausgehenden Anruf verwendet werden kann und einen Zeiger auf die Objektschnittstelle **ITBasicCallControl** zurückgibt.

```
HRESULT
CreateCall (
BSTR *pDialableAddress,
Long 1AddressType,
Long 1MediaTypes,
ITBasicCallControl**ppCall
);
```

IEnumAddress

Stellt Aufzählungsmethoden nach COM-Standard für die **ITAddress**-Schnittstelle bereit.

Next

Die Methode **Next** ruft die nächste angegebene Anzahl an Elementen in der Aufzählungsreihenfolge ab.

```
HRESULT
Next(
ULONG celt,
ITAddress**ppElements,
ULONG *pceltFetched
);
```

ITMediaSupport

Die Schnittstelle **ITMediaSupport** stellt Methoden bereit, mit denen eine Anwendung ermitteln kann, welche Medien das zu dieser Schnittstelle gehörende Adressobjekt (Address) unterstützt.

get_MediaTypes

Die Methode get_MediaTypes ruft die an der aktuellen Adresse unterstützten Medientypen ab.

```
HRESULT
get_MediaTypes (
long *plMediaTypes
);
```

Terminal

Das Terminal-Objekt stellt die Quelle eines mit einem Anruf oder einer Kommunikationssitzung verbundenen Medienstroms dar.

Call

Anrufen

Das Anrufobjekt (Call) stellt die Adressenverbindung zwischen der lokalen Adresse und einer oder mehreren anderen Adressen dar.

ITCallInfo

Die Schnittstelle **ITCallInfo** dient zum Abrufen und Festlegen verschiedener Informationen im Zusammenhang mit einem Call-Objekt.

get_Address

Die Methode **get_Address** ruft einen Zeiger auf die Schnittstelle **ITAddress** des Adressobjekts (Address) ab.

```
HRESULT
get_Address (
ITAddress **ppAddress
);
```

get_CallState

Die Methode **get_CallState** ruft einen Zeiger auf den aktuellen Anrufstatus ab, wie beispielsweise CS_IDLE.

```
HRESULT
get_CallState (
CALL_STATE *pCallState
);
```

get_CallInfoString

Die Methode **get_CallInfoString** ruft Anrufinformationen in Form einer Zeichenfolge ab, wie beispielsweise den Anzeigenamen der Adresse.

```
HRESULT
get_CallInfoString(
CALLINFO_STRING CallInfoString,
BSTR *ppCallInfoString
);
```

SetCallInfoBuffer

TAPI 3.0 (Windows 2000) unterstützt diese Funktion nur für inaktive Anrufe mit dem Status IDLE. Dieses Problem wurde in TAPI 3.1 (Windows XP) gelöst. In dieser Version können Anrufdaten für verbundene Anrufe festgelegt werden, indem CIB_CALLDATABUFFER als Parameter **CallInfoBuffer** übergeben wird.

```
HRESULT
SetCallInfoBuffer (
CALLINFO_BUFFERCallInfoBuffer,
DWORD dwSize
BYTE* pCallInfoBuffer
);
```

ITBasicCallControl

Die Schnittstelle **ITBasicCallControl** ermöglicht der Anwendung das Herstellen einer Verbindung zu einem Call-Objekt sowie das Senden von Antworten und die Ausführung grundlegender Telefoniefunktionen.

Connect

Die Methode **Connect** versucht, die Verbindung eines ausgehenden Anrufs abzuschließen.

```
HRESULT
Connect(
VARIANT_BOOL fSync
);
```

Answer

Die Methode **Answer** beantwortet eingehende Anrufe. Diese Methode kann nur erfolgreich ausgeführt werden, wenn der Anrufstatus CS_OFFERING lautet.

```
HRESULT
Answer();
```

Disconnect

Die Methode **Disconnect** trennt die Anrufverbindung. Nach erfolgreichem Abschluss der Methode ändert sich der Anrufstatus in CS_DISCONNECTED.

```
HRESULT
Disconnect(
DISCONNECT_CODE code
);
```

Halten

Die Methode **Hold** kann einen Anruf halten und einen gehaltenen Anruf aktivieren.

```
HRESULT
Hold(
VARIANT_BOOLfHold
);
```

SwapHold

Die Methode **SwapHold** tauscht den aktiven Anruf gegen den angegebenen gehaltenen Anruf.

```
HRESULT
SwapHold(
ITBasicCallControl *pCall
);
```

ParkDirect

Die Methode **ParkDirect** parkt den Anruf an einer angegebenen Adresse.

```
HRESULT
ParkDirect(
BSTR pParkAddress
);
```

Unpark

Die Methode **Unpark** holt einen Anruf aus dem geparkten Status zurück.

```
HRESULT
Unpark();
```

BlindTransfer

Die Methode **BlindTransfer** führt eine nicht überwachte Vermittlung des angegebenen Anrufs an die angegebene Adresse aus.

```
HRESULT BlindTransfer(
BSTR pDestAddress
);
```

Transfer

Die Methode **Transfer** vermittelt den aktuellen Anruf an die Zieladresse.

```
HRESULT Transfer(
ITBasicCallControl*pCall,
VARIANT_BOOL fSync
);
```

Finish

Die Methode **Finish** wird bei einem Rückfrageruf zum Fertigstellen eines Konferenzgesprächs oder einer Vermittlung aufgerufen.

```
HRESULT Finish(
FINISH_MODE finishMode
);
```

Conference

Die Methode **Conference** schaltet einen Rückfrageruf für das Konferenzgespräch, an dem der aktuelle Anruf teilnimmt.

```
HRESULT Conference(
ITBasicCallControl *pCall,
VARIANT_BOOL fSync
);
```

RemoveFromConference

Die Methode **RemoveFromConference** entfernt den Anruf ggf. aus einem Konferenzgespräch.

```
HRESULT RemoveFromConference();
```

ITCallStateEvent

Die Schnittstelle **ITCallStateEvent** umfasst Methoden zum Abrufen der Beschreibung von Anrufstatusereignissen.

get_Cause

Die Methode **get_Cause** ruft die mit diesem Ereignis verbundene Ursache ab.

```
HRESULT
get_Cause (
CALL_STATE_EVENT_CAUSE *pCEC
);
```

get_State

Die Methode **get_State** ruft Informationen zum neuen Anrufstatus ab.

```
HRESULT
get_State (
CALL_STATE *pCallState
);
```

get_Call

Die Methode **get_Call** ruft einen Zeiger auf die Anrufinformationsschnittstelle ab, an der das Ereignis eingetreten ist.

```
HRESULT
get_Call
ITCallInfo**ppCallInfo
);
```

ITCallNotificationEvent

Die Schnittstelle **ITCallNotificationEvent** umfasst Methoden zum Abrufen der Beschreibung von Anrufereignissen.

get_Call

Die Methode **get_Call** gibt die **ITCallInfo**-Schnittstelle zurück, an der das Anrufereignis eingetreten ist.

```
HRESULT
get_Call
ITCallInfo**ppCall
);
```

ITCallInfoChangeEvent

Die Schnittstelle **ITCallInfoChangeEvent** umfasst Methoden zum Abrufen der Beschreibung von Ereignissen, wenn sich Anrufinformationen ändern.

get_Call

Die Methode **get_Call** gibt die **ITCallInfo**-Schnittstelle zurück, an der sich Anrufinformationen geändert haben.

```
HRESULT
get_Call
ITCallInfo**ppCall
);
```

CallHub

Das Call Hub-Objekt (CallHub) stellt Methoden zum Abrufen von Informationen zu Teilnehmern an einem Konferenzgespräch bereit. Call Hubs werden von IP Office nicht unterstützt. Call Hub-Ereignisse können empfangen werden, sollten aber ignoriert werden.

TAPI 3-Aufzählungstypen

CALL_STATE

CALL_STATE wird von den Methoden **ITCallInfo::get_CallState** und **ITCallStateEvent::get_State** verwendet.

Element	Wert	Beschreibung
CS_IDLE	0	Der Anruf wurde erstellt, **Connect** wurde jedoch noch nicht aufgerufen. Ein Anruf kann nie in diesen Status übergehen.
CS_INPROGRESS	1	**Connect** wurde aufgerufen und der Dienstanbieter versucht, eine Verbindung herzustellen. Dieser Status ist nur für ausgehende Anrufe gültig. Diese Meldung ist optional, da möglicherweise der direkte Übergang zum verbundenen Status erfolgt.
CS_CONNECTED	2	Der Anruf wurde verbunden und das Gespräch kann begonnen werden.
CS_DISCONNECTED	3	Die Verbindung wurde getrennt. Es gibt verschiedene Ursachen für die Trennung einer Verbindung. Siehe folgende Tabelle der gültigen Anrufstatusübergänge.
CS_OFFERING	4	Ein neuer Anruf liegt vor und wartet an einer Anwendung. Wenn die Anwendung über Besitzerrechte für den Anruf verfügt, kann sie entweder **Answer** oder **Disconnect** aufrufen, während der Anruf sich im wartenden Status befindet.
CS_HOLD	5	Der Anruf wird gehalten.
CS_QUEUED	6	Der Anruf wird in die Warteschlange gestellt.

CALLINFO_STRING

CALLINFO_STRING wird von den **ITCallInfo**-Methoden verwendet, die zum Abrufen und Festlegen von Anrufinformationen mit Hilfe von Zeichenfolgen dienen.

Element	Wert	Beschreibung
CIS_CALLERIDNAME	0	Der Name des Anrufers.
CIS_CALLERIDNUMBER	1	Die Nummer des Anrufers.
CIS_CALLEDIDNAME	2	Der Name des angerufenen Standorts.
CIS_CALLEDIDNUMBER	3	Die Nummer des angerufenen Standorts.
CIS_CONNECTEDIDNAME	4	Der Name des verbundenen Standorts.
CIS_CONNECTEDIDNUMBER	5	Die Nummer des verbundenen Standorts.
CIS_REDIRECTIONIDNAME	6	Der Name des Standorts, an den der Anruf umgeleitet wurde.
CIS_REDIRECTIONIDNUMBER	7	Die Nummer des Standorts, an den der Anruf umgeleitet wurde.
CIS_REDIRECTINGIDNAME	8	Der Name des Standorts, der den Anruf umgeleitet hat.
CIS_REDIRECTINGIDNUMBER	9	Die Nummer des Standorts, der den Anruf umgeleitet hat.
CIS_CALLEDPARTYFRIENDLYNAME	10	Der angezeigte Name der angerufenen Adresse.
CIS_COMMENT	11	Ein Kommentar zu dem Anruf von der Anwendung, von der der Anruf ausging.
CIS_DISPLAYABLEADDRESS	12	Die Anzeigeversion der angerufenen Adresse bzw. der Adresse des Anrufers.
CIS_CALLINGPARTYID	13	Die ID des Anrufers.

DISCONNECT_CODE

DISCONNECT_CODE wird von der Methode **ITBasicCallControl::Disconnect** verwendet.

Element	Wert	Beschreibung
DC_NORMAL	0	Die Verbindung wird unter normalen Bedingungen getrennt, d.h. das Gespräch wurde beendet.
DC_NOANSWER	1	Die Verbindung wird getrennt, da der Anruf nicht beantwortet wurde. (In einer Anwendung kann beispielsweise eingestellt werden, wie lange auf eine Antwort gewartet wird. Wenn keine Antwort erfolgt, ruft die Anwendung **Disconnect** mit dem Code NOANSWER auf.)
DC_REJECTED	2	Der wartende Anruf wurde abgelehnt.

CALL_STATE_EVENT_CAUSE

CALL_STATE_EVENT_CAUSE wird von der Methode **ITCallStateEvent::get_Cause** zurückgegeben.

Element	Wert	Beschreibung
CEC_NONE	0	Es ist kein Anrufereignis aufgetreten.
CEC_DISCONNECT_NORMAL	1	Die Verbindung wurde unter normalen Bedingungen getrennt (d.h. nach Beendung des Gesprächs).
CEC_DISCONNECT_BUSY	2	Ein ausgehender Anruf konnte nicht durchgestellt werden, da die angerufene Nummer besetzt war.
CEC_DISCONNECT_BADADDRESS	3	Ein ausgehender Anruf ist fehlgeschlagen, da die Zieladresse fehlerhaft war.
CEC_DISCONNECT_NOANSWER	4	Ein ausgehender Anruf ist fehlgeschlagen, da die angerufene Nummer nicht geantwortet hat.
CEC_DISCONNECT_CANCELLED	5	Ein ausgehender Anruf ist fehlgeschlagen, da der Anrufer die Verbindung getrennt hat.
CEC_DISCONNECT_REJECTED	6	Der ausgehende Anruf wurde vom angerufenen Teilnehmer nicht angenommen.
CEC_DISCONNECT_FAILED	7	Die Verbindung konnte aufgrund anderer Ursachen nicht hergestellt werden.

IP Office Media Service Provider

Info zum MSP

Der IP Office Media Service Provider ist doppelt nützlich. Er bietet Media-Streaming-Funktionen, mit denen TAPI 3-Anwendungen Sprachdaten für Anrufe in spezifischen Typen von Benutzerleitungen senden und empfangen können. Außerdem ermöglicht er Anwendungen den Zugriff auf gerätespezifische Funktionen von IP Office.

Verwenden des MSP

Die Schnittstellen des Media Service Provider sind in den MSDN-Bibliotheken dokumentiert. Das Beispiel **DevSpice** auf der SDK-CD gibt einen Einblick in die Verwendung des MSP für Media-Streaming- und gerätespezifische Funktionen. Der MSP ist für jede TAPI-Adresse verfügbar, die in Ihrer TAPI 3-Anwendung angezeigt wird. Die Media-Streaming-Funktionen stehen nur für Adressen zur Verfügung, die als Wave-Benutzer gekennzeichnet sind. WAVE-Benutzer werden an ihrem Benutzernamen erkannt, der mit "TAPI" beginnt. (Beispielsweise "TAPI:201") Sie können beliebig viele WAVE-Benutzer erstellen. Für jeden Wave-Benutzer ist jedoch eine Wave-Treiber-Lizenz erforderlich, um das Senden von Medienströmen an diesen Benutzer zu ermöglichen.

Verwenden der gerätespezifischen Schnittstellen

Die gerätespezifischen Schnittstellen werden im Adressobjekt (**Address**) und im Anrufobjekt (**Call**) des MSP implementiert. In TAPI 3.0 werden Abfragen für Schnittstellen, die nicht erkannt werden, an den MSP übergeben. Wenn Sie beispielsweise über einen Zeiger auf eine Schnittstelle **ITAddress** verfügen, können Sie mit Hilfe von **QueryInterface** einen Zeiger auf die Schnittstelle **ITDivert** abrufen. Siehe folgenden Code aus dem Beispiel **DevSpice**:

```
ITDivert* pDivert = NULL;
if( SUCCEEDED( gpAddress->QueryInterface( IID_ITDIVERT,
(void**)&pDivert)))
{
DWORD dwDivertSettings = 0;
if( FAILED( pDivert-> GetDivertSettings(
&dwDivertSettings)))
{
```

Folgende Schnittstellen sind vom Adressobjekt (Address) verfügbar:

- ITACDAgent
- ITDivert
- ITGroup

Folgende Schnittstelle ist vom Anrufobjekt verfügbar:

- ITPlay

Des Weiteren fungiert das Adressobjekt (Address) als Verbindungspunkt-Container für private Ereignisse in IP Office. Die Verbindungspunktschnittstelle mit dem Namen **IPOfficePrivateEvent** ist in der Datei **interfaces.h** des Beispiels **DevSpice** verfügbar. Nähere Informationen zu diesen Schnittstellen finden Sie im Folgenden.

ITACDAgent

IsLoggedIn(void)	Gibt S_TRUE zurück, wenn der Benutzer angemeldet ist, und S_FALSE, wenn der Benutzer abgemeldet ist.
LogOut(void)	Meldet den Benutzer bei dieser Leitung ab. Für den Benutzer muss in Manager die Einstellung **Login erzwingen** aktiviert sein.
Login(BSTR extn)	Meldet den Benutzer an der gegebenen Nebenstelle an.
CallListen(BSTR extn)	Hört den Anruf an der gegebenen Nebenstelle ab. Für den Benutzer muss in Manager die Einstellung **Aufschalten erlaubt** aktiviert sein.
Intrude(BSTR extn)	Schaltet den aktuellen Benutzer in den Anruf an der gegebenen Nebenstelle. Für den Benutzer muss in Manager die Einstellung **Aufschalten erlaubt** aktiviert sein.
SetAccountCode(BSTR extn)	Legt den Verrechnungscode für den aktuellen Anruf fest.

ITGroup

Diese Schnittstelle bietet Funktionen zur Aktivierung bzw. Deaktivierung von Gruppenmitgliedschaften eines Benutzers sowie zum Abfangen von Anrufen an anderen Telefongeräten der Gruppe.

PickupAny(void)	Entspricht der Ausführung des Funktionscodes **CallPickupAny** am Benutzerendgerät. Nähere Informationen finden Sie unter **Manager**.
PickupGroup(void)	Entspricht der Ausführung des Funktionscodes **CallPickupGroup** am Benutzerendgerät.
PickupExtn(BSTR extn)	Entspricht der Ausführung des Funktionscodes **CallPickupExtn** am Benutzerendgerät.
PickupMembers(BSTR extn)	Entspricht der Ausführung des Funktionscodes **CallPickupMembers** am Benutzerendgerät.
Enable(BSTR groupextn)	Aktiviert die Mitgliedschaft des Benutzers bei der gegebenen Gruppe. Wenn **groupextn** eine leere Zeichenfolge enthält, werden alle vorhandenen Gruppenmitgliedschaften des Benutzers aktiviert.
Disable(BSTR groupextn)	Deaktiviert die Mitgliedschaft des Benutzers bei der gegebenen Gruppe. Wenn **groupextn** eine leere Zeichenfolge enthält, werden alle vorhandenen Gruppenmitgliedschaften des Benutzers deaktiviert.

ITDivert

Diese Schnittstelle bietet Funktionen zum Abrufen und Festlegen der Weiterleitungseinstellungen für die Adresse.

GetDivertAllDestination(BSTR* pDestination)	Ruft das aktuelle Ziel für **Alle weiterleiten** ab und gibt das Ergebnis im Wert **pDestination** zurück.
SetDivertAllDestination(BSTR dest)	
GetDivertSettings(DWORD* pdwDivertSets)	Diese Funktion legt Bit für den Wert DWORD fest, auf den **pdwDivertSets** verweist. So wird gekennzeichnet, welche Weiterleitungseinstellungen zurzeit aktiv sind. Die Bit werden mit Hilfe des Enumerators IP_OFFICE_DIVERT_SETTINGS (weiter unten beschrieben) definiert.
SetForwardAll(VARIANT_BOOL bOn)	Aktiviert/deaktiviert die Einstellung **Alle weiterleiten** für diesen Benutzer.
SetForwardBusy(VARIANT_BOOL bOn)	Aktiviert/deaktiviert die Einstellung **Rufweiterleitung bei Besetzt** für diesen Benutzer.
SetForwardNoAnswer(VARIANT_BOOL bOn)	Aktiviert/deaktiviert die Einstellung **Rufweiterleitung nach Zeit** für diesen Benutzer.
SetDoNotDisturb(VARIANT_BOOL bOn)	Aktiviert/deaktiviert die Einstellung **Nicht stören** für diesen Benutzer.

Der Enumerator IP_OFFICE_DIVERT_SETTINGS wird wie folgt definiert:

```
typedef enum
{
IPOFF_FWDALL = 0x01,
IPOFF_FWDBUSY = 0x02,
IPOFF_NOANSWER = 0x04,
IPOFF_DND = 0x08,
IPOFF_DESTINATION = 0x10
} IP_OFFICE_DIVERT_SETTINGS;
```

Das Ergebnis 14 (0xe) von **GetDivertSettings** bedeutet somit, dass für den Benutzer die Einstellungen **Rufweiterleitung bei Besetzt**, **Rufweiterleitung nach Zeit** und **Nicht stören** aktiviert sind. Der Wert IPOFF_DESTINATION wird nicht von **GetDivertSettings** verwendet, sondern lediglich von der Funktion **Fire_DivertSettingsChanged** an der Schnittstelle **IPOfficePrivateEvents**.

ITPlay

Die Schnittstelle **ITPlay** ist im Objekt **MSP Call** implementiert. Sie ermöglicht das Aufzeichnen und Abspielen von Wave-Dateien.

StartPlay(BSTR FileName)	**FileName** muss den vollständigen Pfad der Wave-Datei wiedergeben, die abgespielt werden soll.
StopPlay()	
StartRecord(BSTR FileName)	**FileName** muss den vollständigen Pfad der Wave-Datei für die Aufzeichnung wiedergeben.
StopRecord()	

Das Abspielen und Aufzeichnen kann zu einem beliebigen Zeitpunkt des Anrufs gestoppt und gestartet werden. Für die Aufzeichnung wird jedoch nur eine Datei pro Anruf verwendet. Wenn die Aufzeichnung gestoppt und erneut gestartet wird, werden die Daten an diese Datei angehängt. Abspiel- und Aufzeichnungsvorgänge sollten nicht gleichzeitig ausgeführt werden. Wenn dies erforderlich ist, können Sie Endgeräte in den Anruf schalten, über die Audiodaten aus einer Datei bereitgestellt bzw. Audiodaten in einer Datei aufgezeichnet werden. In TAPI 3.1 wird dieser Vorgang mit Hilfe von Datei-Streaming-Endgeräten vereinfacht.

IPOfficePrivateEvents

Hierbei handelt es sich um eine Verbindungspunktschnittstelle, die vom MSP für Ereignisberichte verwendet wird. Informationen zur Registrierung für und zur Handhabung von privaten Ereignissen finden Sie im Beispiel **DevSpice**.

OnUserLogin(void)	Ausgelöst, wenn ein Agent (ein Benutzer, für den in Manager die Einstellung **Login erzwingen** aktiviert ist) sich anmeldet.
OnUserLogout(void)	Ausgelöst, wenn ein Agent sich abmeldet.
OnDivertSettingsChanged(DWORD dwDivertSettings)	Ausgelöst, wenn der Benutzer eine seiner Umleitungseinstellungen (wie **Nicht stören** oder **Rufweiterleitung bei Besetzt**) oder das Ziel für **Alle weiterleiten** ändert. Die Bit in der Variablen **dwDivertSettings** werden unter Verwendung von IP_OFFICE_DIVERT_SETTINGS (oben beschrieben) festgelegt.
OnGroupChanged(DWORD dwGroupCount)	Ausgelöst, wenn der Benutzer eine Gruppenmitgliedschaft aktiviert oder deaktiviert. Mit dem Wert **dwGroupCount** wird die Anzahl der Gruppen angegeben, für die der Benutzer als Mitglied aktiviert ist.
OnVoiceMail(DWORD dwNumMessages)	Ausgelöst, wenn sich die Anzahl der neuen VoiceMail-Nachrichten für einen Benutzer ändert. Der neue Wert wird mit dem Parameter **dwNumMessages** ausgegeben.

Verwenden der Media-Streaming-Funktionen des MSP

Der IP Office MSP ermöglicht die Handhabung von Media-Streaming an jeden beliebigen Wave-Benutzer. Hierzu müssen Sie Endgeräte für die Datenströme schalten, die der MSP für einen Anruf bereitstellt. Nähere Informationen zur Vorgehensweise finden Sie im MSDN (Microsoft Developer Network) und ein Beispiel im Beispiel **DevSpice**. Beachten Sie, dass IP Office-Datenströme bidirektional sind und nur ein Datenstrom pro Anruf vorhanden ist. Dies bedeutet, dass jeweils ein Sende- und ein Empfangsgerät für denselben Datenstrom akzeptiert werden.

Der MSP beinhaltet die Funktionen des IP Office-Wave-Treibers. Der IP Office-Wave-Treiber muss auf jedem Gerät installiert sein, mit dem Medienströme an Benutzer gesendet werden sollen. Wenn Sie kein Media-Streaming durchführen, sondern lediglich Wave-Benutzer mit TAPI überwachen möchten, stellen Sie sicher, dass der Wave-Treiber nicht installiert ist. Andernfalls nutzen Sie für jede geöffnete Wave-Benutzerleitung unnötig eine Wave-Lizenz.

Index

A

Abgemeldet 10
Address 40
Anmeldeprotokoll 10
Anrufinformationen 48
Anrufstatus
  Ereignisstatus 49
Anrufstatus 47
Aufschalten 13

B

Benutzung
  Geräteschnittstellen 51
  Media-Streaming 55
  MSP 51

C

Call 42
Call Hub 46

D

disconnect_code 48

E

Einstellungen für die Weiterleitung 12

G

Geräteschnittstellen 51
Gruppe aktivieren/deaktivieren 12

I

Installieren
  Lizenzen 3
  Treiber 3

K

Konfigurieren
  IP Office 5
  Treiber 3

M

Media-Streaming 55
Mithören 13
MSP 51

N

Nachricht wartet (Leuchte) 11

P

Private Ereignisse 54

T

TAPI
  Ereignismeldungen 37
  Funktionen 7
  TAPI 2 5
  TAPI 3 5
terminal 42

U

Umleitungsziel 11

V

Verweise 1

Anmerkungen oder Vorschläge bezüglich dieses Dokuments können an "wgctechpubs@avaya.com" gesendet werden.

Avaya
Unit 1, Sterling Court
15 - 21 Mundells
Welwyn Garden City
Hertfordshire
AL7 1LZ
England

Tel: +44 (0) 1707 392200
Fax: +44 (0) 1707 376933

Web: http://www.avaya.com/ipoffice/knowledgebase